

# Software Design Document

Group No. 28 : Prevention of Cyber Troll &  
Sarcasm System on Social Networking using  
Machine Learning with Bilingual Analytics

Tejas Karia, Priya Mane, Jeet Mehta, Pratik Merchant

October 2020

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Design Overview . . . . .	3
1.2	Requirements Traceability Matrix . . . . .	3
<b>2</b>	<b>System Architectural Design</b>	<b>4</b>
2.1	Chosen System Architecture . . . . .	4
2.2	Discussion of Alternative Designs . . . . .	4
2.3	System Interface Description . . . . .	4
2.3.1	User Interfaces . . . . .	4
2.3.2	Hardware Interfaces . . . . .	5
2.3.3	Software Interfaces . . . . .	5
2.3.4	Communications Protocols . . . . .	5
<b>3</b>	<b>Detailed Description Of Components</b>	<b>5</b>
3.1	Component 1: User . . . . .	5
3.2	Component 2: API . . . . .	6
3.3	Component 3: Sentiment Analysis . . . . .	6
3.4	Component 4: Automating Replying/Blocking Process . . . . .	6
3.5	Component 5: Database . . . . .	7
<b>4</b>	<b>User Interface Design</b>	<b>7</b>
4.1	Description of the User Interface . . . . .	7
4.1.1	Screen Images . . . . .	7
4.1.2	Objects and Actions . . . . .	10

<b>5</b>	<b>System Architecture</b>	<b>11</b>
<b>6</b>	<b>Data Flow Specifications</b>	<b>12</b>
6.1	Level 0 DFD with description: . . . . .	12
6.2	Level 1 DFD with description: . . . . .	13

# 1 Introduction

## 1.1 Design Overview

This project will help to deal with online social media hate speech and automate the process of blocking such malicious accounts. The current process for the social media platforms are manual and there are no automated processes. Since the process is manual it becomes very difficult to keep track of such users who are habitual offenders. There are several categories of cyber hate and each of these are interpreted differently. The project has broken this down to mainly 2 categories: offensive and sarcastic depending on the sentiment bilingual sentiment analysis on Hinglish comments. Our main target for developing this tool is to empower influencers who do not have the time to tackle the hate speech and thus they have to keep a social media manager who has to manually delete such malicious comments. Primarily, all the comments will be retrieved from the user created posts and then those will be classified using sentiment analysis. An automated response will be generated to the comments. The product will be designed using Bottom-up Object Oriented Design approach. All these separate modules will be generated first and then these subsystems will be clubbed to form our main system.

## 1.2 Requirements Traceability Matrix

	User	API/OAuth	Sentiment Analysis	Automation	Database
User Login	X				X
User Authentication for Social Media Platform.	X	X			
Retrieval & Classification of Comments	X	X	X		
Generate responses for comments received.			X	X	
Block spam users and report/delete negative comments		X	X	X	

## 2 System Architectural Design

### 2.1 Chosen System Architecture

#### **Data-Centered Architecture:**

The main component of the web application is the data pulled in from the various APIs. The secondary component consists of the modules processing the data. The user is a passive spectator and makes few decisions like assisting the auto-reply and blocking users. Hence, the architecture chosen is Data-Centered architecture. The major risk associated with the application would have been regarding user credentials security, but this risk can be eliminated by using OAuth provided by the social media platform itself. Another risk would be of misidentifying spam users or negative comments because of algorithmic flaws, to eliminate this risk, the application will present a preview for all the actions it will be performing to seek rectification from the user.

### 2.2 Discussion of Alternative Designs

The alternate application architecture can be Data-Centered Blackboard type. In Blackboard Architecture Style, the data stored is active and its clients are passive. Therefore, the logical flow is determined by the current data status in the data store. The Knowledge Sources (KS) is the data from APIs. Blackboard Data Structure is the user console/dashboard component. Control is the user itself. This architecture was not chosen because the application is not entirely passive as the user has to interact with the application for approving changes like automated replies and user blocking, etc.

### 2.3 System Interface Description

#### 2.3.1 User Interfaces

Virtual Social Media Manager is a web application that allows the user to analyse their social media comments and eliminate negative content by features like blocking users posting negative remarks and reporting/blocking hate comments.

- The web application consists of a first page i.e the application home, from where the user can navigate to login/sign up pages.
- On the sign up page, the user has to enter a few basic details and choose a safe password.
- On the Login page, the user has to enter his/her email ID and password to login.
- After Login, the user will be redirected to his home page/Dashboard containing his/her details. From here the user can navigate to various panels for each social media platform.
- On each panel, the user will be redirected to an OAuth page for that particular platform. The user has to authenticate himself and then he/she will be

redirected to the panel page for that social media platform.

- On the panel page, all the user's posts are retrieved and the comments are analysed and displayed in a tabular form. The user can also select to view content specific review.

### 2.3.2 Hardware Interfaces

There are no specific hardware requirements for using the website as it would be hosted over the internet.

### 2.3.3 Software Interfaces

Operating System: The application would be compatible with all the operating systems which have the compatible browsers installed. The website would be accessible on various browsers such as Google Chrome, Mozilla Firefox and Microsoft Edge to name a few.

Database: For storing user data we will be making use of MySQL.

### 2.3.4 Communications Protocols

For uploading data to the database and retrieving the data from the database over the internet, the relevant TCP protocols will be used. OAuth will be used for establishing communication with the APIs.

## 3 Detailed Description Of Components

### 3.1 Component 1: User

Responsibilities	1. Create an account if not already created. 2. Login
Constraints	Password string should match with that in the database.
Composition	Frontend: HTML5, CSS3. Django Framework. Backend: MySQL
Interaction	Users log into the application. Provides the credential detail for OAuth. The user can review the changes like automated replies and block the users with malicious intent.
Resources	Developer

### 3.2 Component 2: API

Responsibilities	The responsibilities of API to provide broad access to public data that users have chosen to share. It also allows users to manage their own non-public information and provide information to developers.
Constraints	Requires an API key without which the access is not granted to the developer.
Composition	JSON
Interaction	This helps us in retrieval of comments and thus we can further use this data and implement various algorithms to calculate its sentiment value.
Resources	Developers should have appropriate API keys which are given upon after applying formally with all the verifications of the developer account done by the backend team.

### 3.3 Component 3: Sentiment Analysis

Responsibilities	Comments retrieved with the help of API will be then further classified into sarcasm or offensive according to the sentiment value it possesses.
Constraints	Requires a huge dataset. The model must perform with an acceptable accuracy.
Composition	Python and libraries such as NLP, <u>scikitlearn</u> , and NLTK.
Interaction	It interacts with the API and data obtained by the developer. Once the data is classified into sarcasm or offensive with the help of a value predefined for it then only the sentiment analysis is considered to be completed.
Resources	Algorithms and Google Colab for training the model.

### 3.4 Component 4: Automating Replying/Blocking Process

Responsibilities	Detect the hate comments which were classified as mentioned earlier and then block that account.
Constraints	User IDs identified to be blocked must be correct and sentiment value assessed must be accurate.
Composition	Will be coded such that the user will be blocked automatically and if the comment is appropriate then only it will be displayed on the post.
Interaction	The sentiment value calculated should help with identifying whether a particular account is to be blocked or not. An automated reply will be generated for appropriate comments.
Resources	Appropriate API endpoints.

### 3.5 Component 5: Database

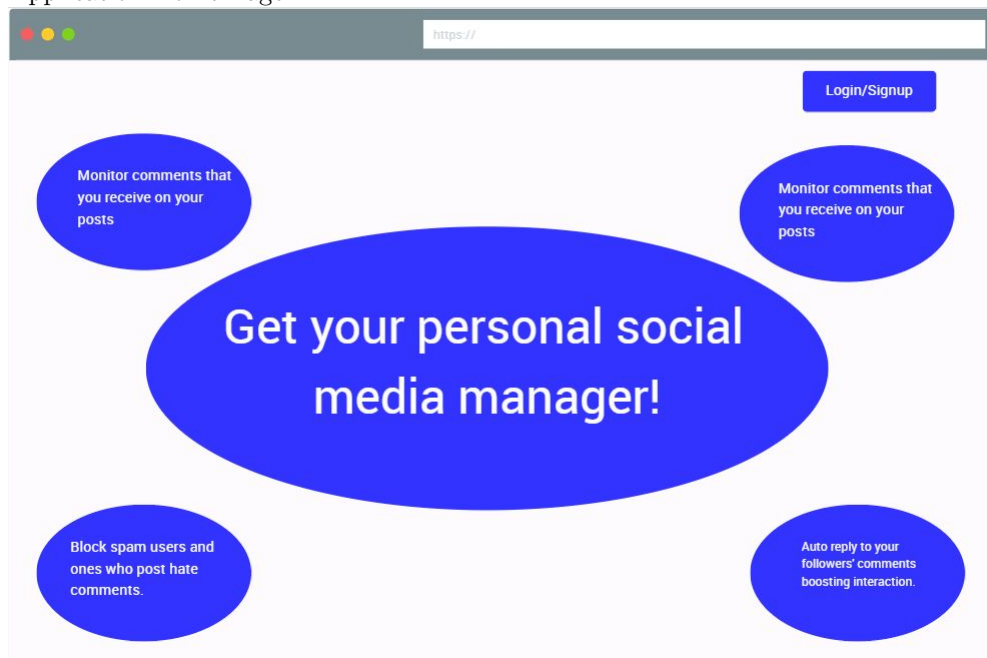
Responsibilities	Database stores information such as username & password.
Constraints	Erroneous data must be avoided .
Composition	MySQL.
Interaction	The application interacts with the database to ensure that credentials entered by the user is correct.
Resources	MySQL.

## 4 User Interface Design

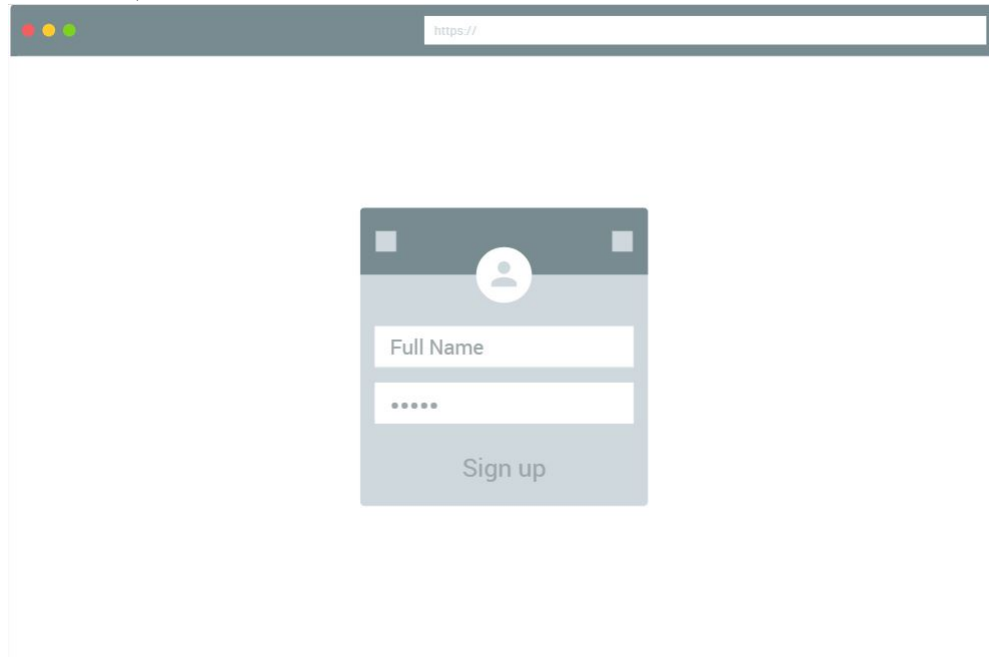
### 4.1 Description of the User Interface

#### 4.1.1 Screen Images

Application Home Page:

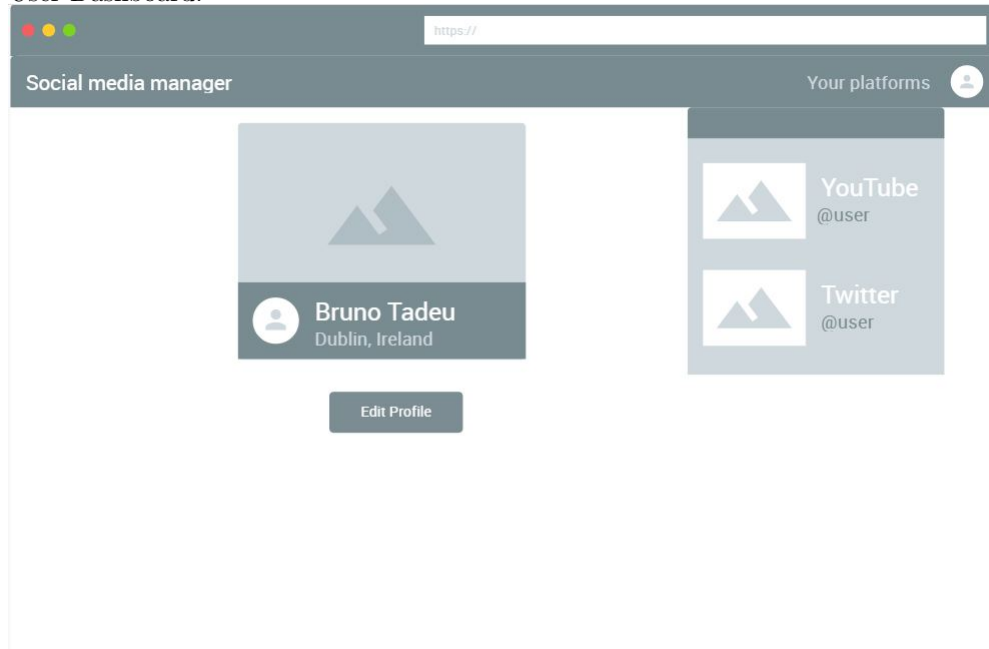


### User Login/Sign Up:



A web browser window showing a sign-up form. The browser's address bar contains "https://". The form is centered on the page and has a light gray background. It features a dark gray header bar with a white user icon in the center. Below the header, there are two input fields: the first is labeled "Full Name" and the second contains six dots, indicating a password field. At the bottom of the form is a "Sign up" button.

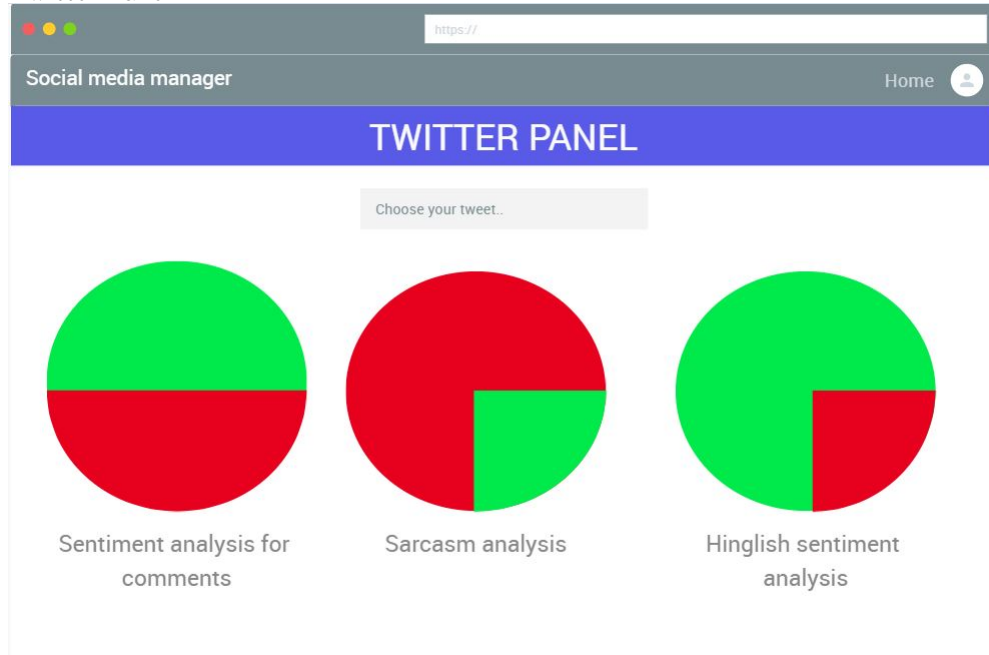
### User Dashboard:



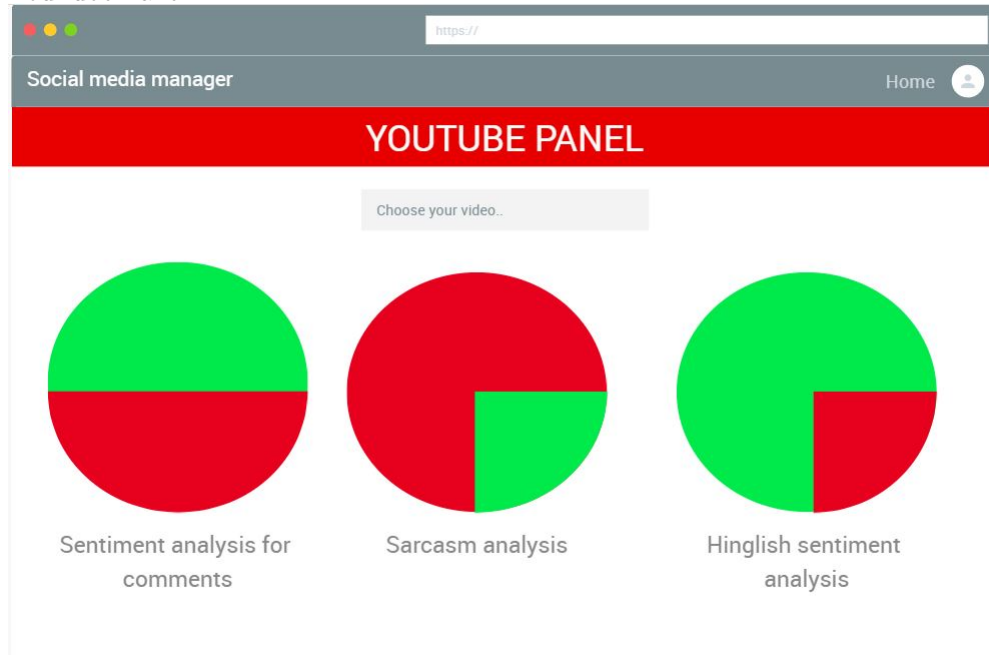
A web browser window showing a user dashboard. The browser's address bar contains "https://". The dashboard has a dark gray header bar with the text "Social media manager" on the left and "Your platforms" with a user icon on the right. The main content area is divided into two sections. On the left, there is a large gray box with a mountain icon and a dark gray footer containing a user icon, the name "Bruno Tadeu", and the location "Dublin, Ireland". Below this box is an "Edit Profile" button. On the right, there is a vertical list of social media platforms. The first entry is "YouTube @user" with a mountain icon. The second entry is "Twitter @user" with a mountain icon.



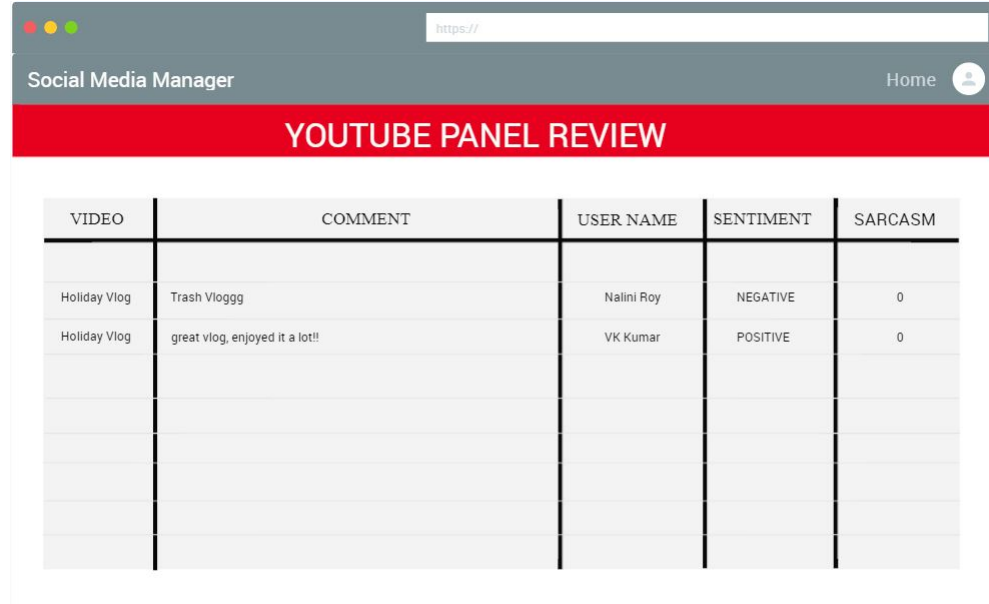
Twitter Panel:



YouTube Panel:



Detailed Review for YouTube:



VIDEO	COMMENT	USER NAME	SENTIMENT	SARCASM
Holiday Vlog	Trash Vloggg	Nalini Roy	NEGATIVE	0
Holiday Vlog	great vlog, enjoyed it a lot!!	VK Kumar	POSITIVE	0

#### 4.1.2 Objects and Actions

##### ●Sign Up and Login:

Sign Up and Login will be displayed in a common screen and the user has to enter only his/her email and password to create a new account as well as for login. The user can then access his account and edit all other details.

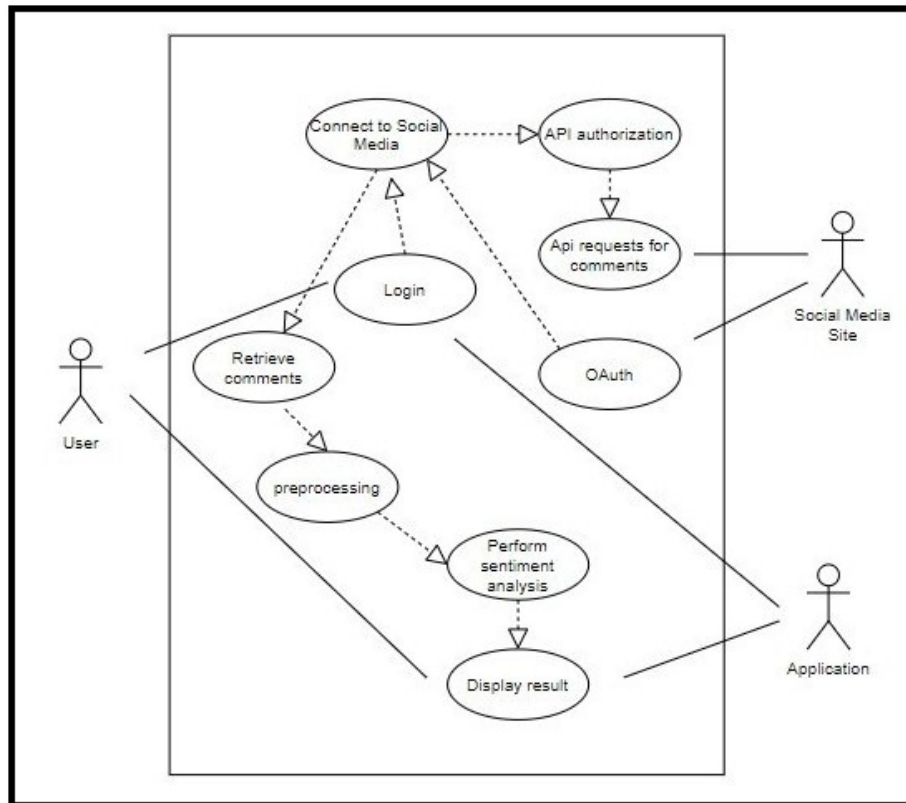
##### ●User Dashboard:

The user can view his details and edit his profile. From here he can navigate to various platform specific panels for viewing the analysis of their posts.

##### ●Platform Specific Panel:

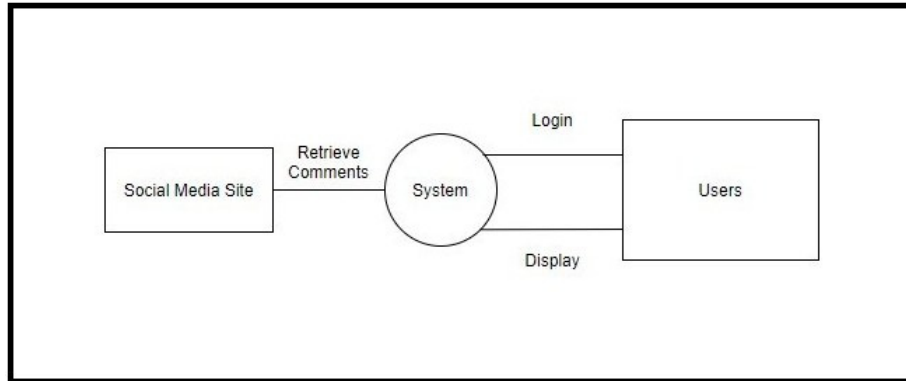
If the user is not logged in and not authenticated the web application, he will be redirected to the OAuth page for the specific platform. After a successful authentication, the user will be redirected to the panel page populated with the data and sentiment and sarcasm analysis done on the comments. The panel acts like a platform specific dashboard. The flagged comments will be highlighted to get user approval for reporting/blocking actions to be performed.

## 5 System Architecture



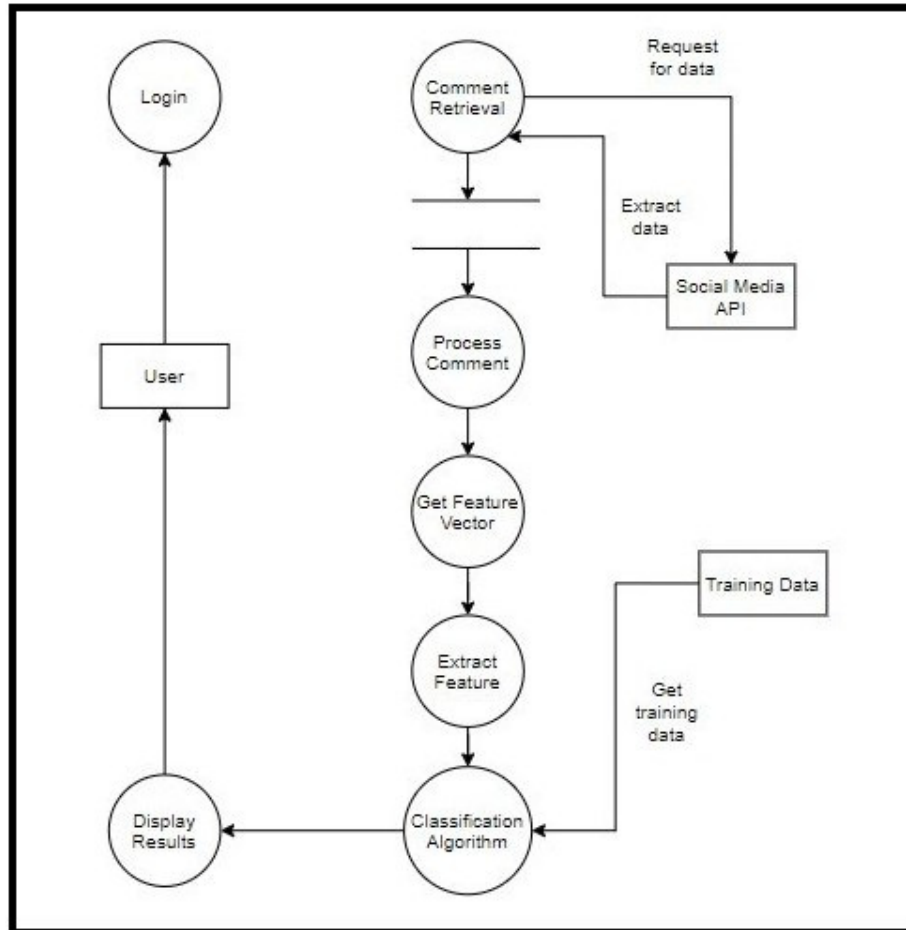
## 6 Data Flow Specifications

### 6.1 Level 0 DFD with description:



Context level data flow diagram depicts the relationship between the system and external data which behaves as source and sink. It is also known as "Level 0 DFD". Under a context diagram, the system interacts with outside agents. The entire system under level 0 DFD acts as a single process and there is no description about internal processing.

## 6.2 Level 1 DFD with description:



The high level data flow diagram shows the division of the system into sub-processes (systems), each describing the data flows with the outside world (agent). It also shows dataflow between each sub-system and performs internal data storage. The purpose of high level is to describe the major processes and their correlations. It can be balanced with its top parent level 0 DFD.