

# Software Requirement Specification Document

---

John Hani, Mohamed Nashaat, Mostafa Ahmed, Zeyad Emad  
Supervised by Dr. Eslam Amer and Eng. Menna Gamil

June 27, 2019

## 1 Introduction

### 1.1 Purpose of this document

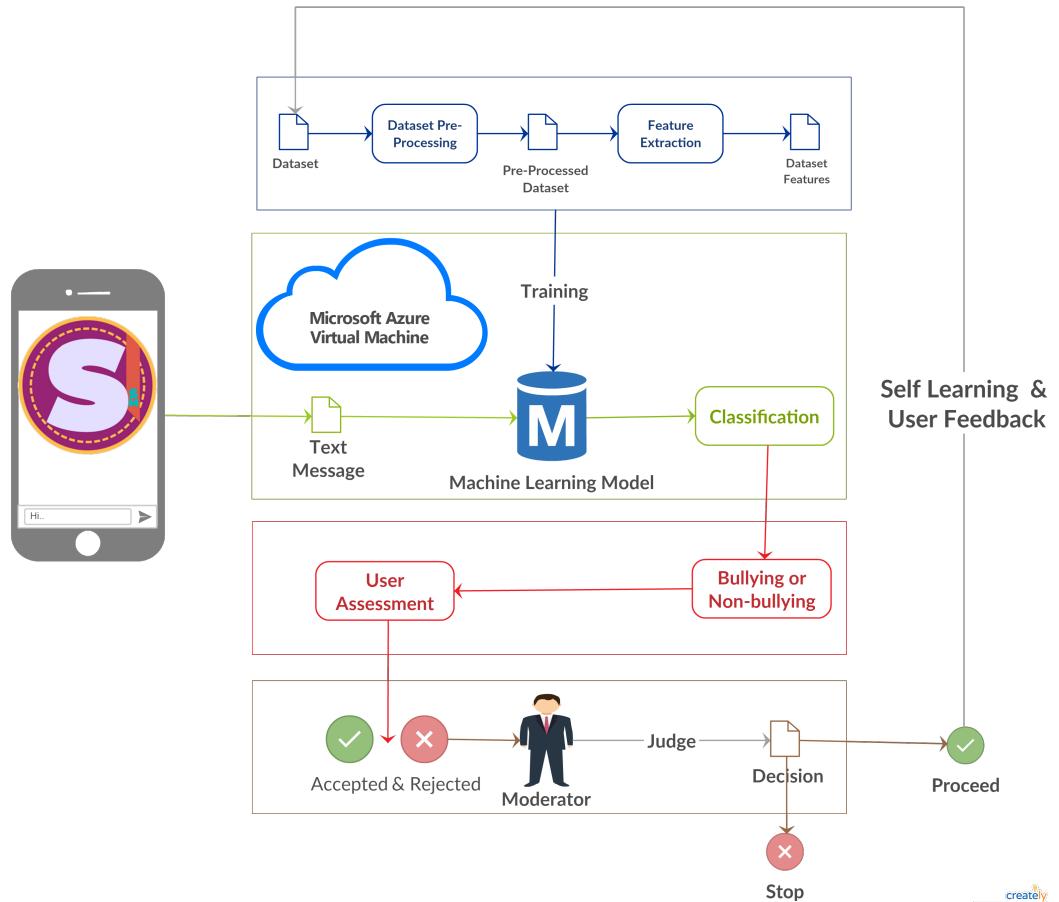
This document serves the purpose of the software requirement specification documentation and to provide a detailed overview of our software product, its parameters and goals and to describe the project's target audience and its user interface, hardware and software requirements.

### 1.2 Scope of this document

It defines how our client, team and audience see the product and its functionality. Our main concern here is that everyone gets the right idea as our functionality as presented to the audience is to develop a system which would be considered as a sub-system on a larger scale to detect and take further actions upon encountering cyberbullying. Our time line of delivery shall be June 2019, while it may cost 0 LE as it is presented for academic purposes, however that may change in the future while taking into consideration the market need. Nonetheless, it helps any designer and developer to assist in software delivery life cycle (SDLC) processes.

### 1.3 Overview

Our Developed application can be part of a larger system in the future



Message: the message will be received to our application from another person.

Smart phones: The application will be on smart phones that have internet connection and minimum RAM 512.

Cloud: The application will send the received message to the cloud .Then the message will go the classifier to detect that there is cyberbullying or not then the cloud will send the output to the application again to rate it.

Dataset: The Dateset size will increase with the help self-learning, as messages of wrong classification will be sent back once more to the dataset with the proper classification for retraining and refining after being revised by a supervisor.

### 1.3.1 User Management

**This Module of the System tends to Manage the Use like Giving access to the Admins and Blocking Some users**

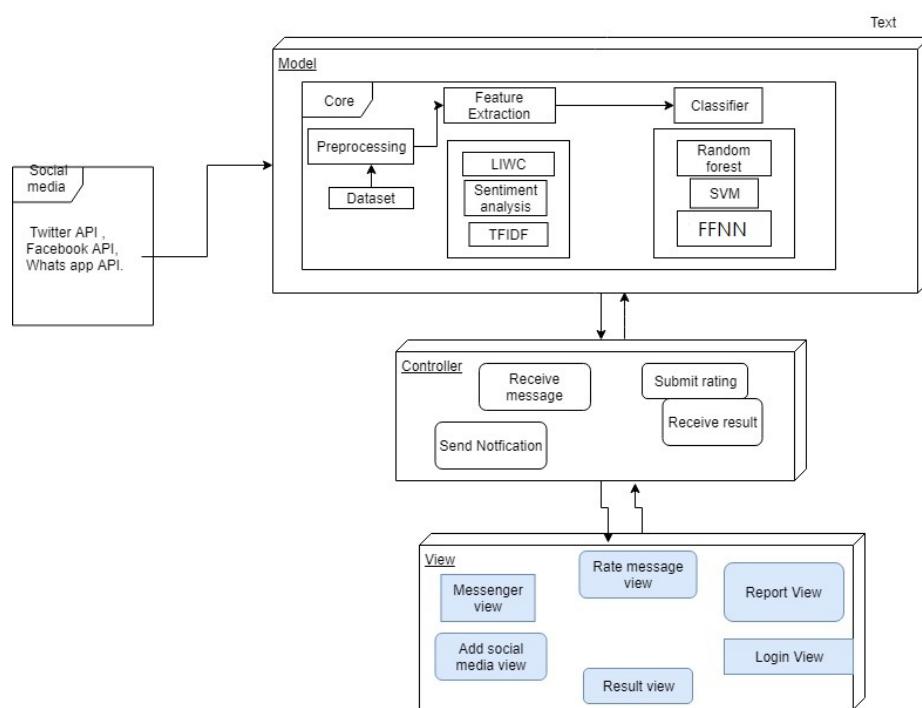


Figure 1: System Architecture

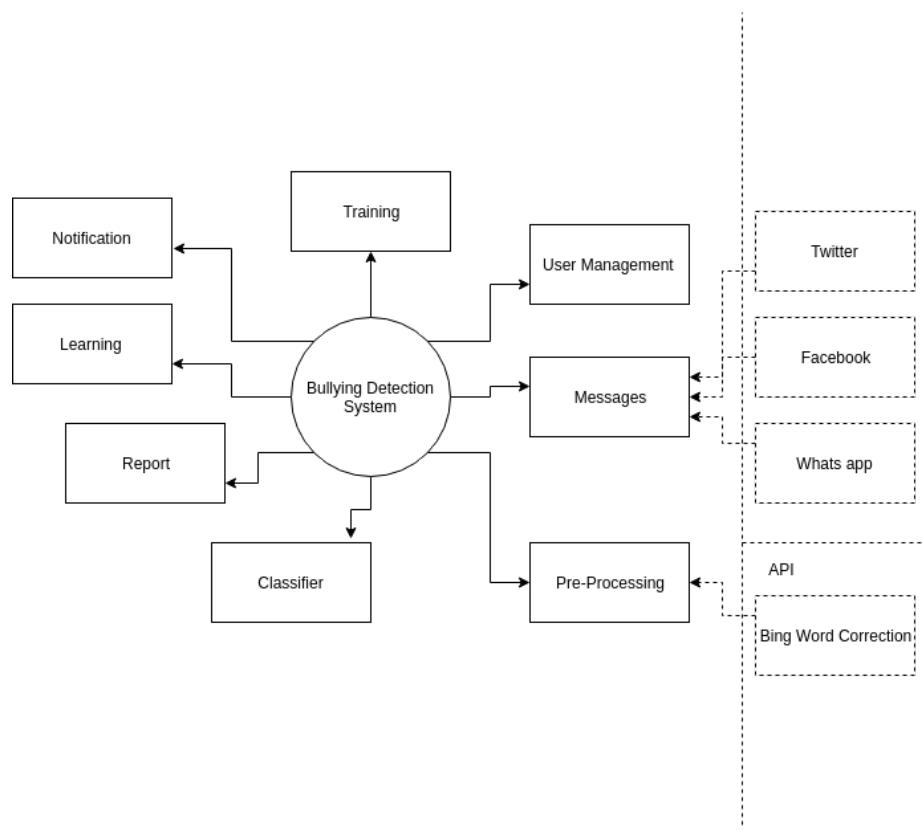


Figure 2: Context Diagram

### **1.3.2 Pre-Processing**

The Pre-Processing Module is the first stage of processing the messages or posts before classification and feature extraction and the pre-processing consists of Tokenization, Stemming, Stop word removal, Lemmatization, Remove the Encoding Parts and the word correction where we use bing word correction API

### **1.3.3 Classifier**

For the Classification we are going to have SVM classifier as our machine learning model, and Feed forward neural network for our deep learning model.

### **1.3.4 Messages**

The User application will send and receive messages from his friends.

### **1.3.5 Report**

Everyday the System is going to produce A detailed Report about the day events like (How many blocked bully messages, Average number of stars of the day, How many wrong classification made based on the user Rating.

### **1.3.6 Self-Learning**

The Self-Learning module is a self learning technique as messages of wrong classification will be sent back once more to the dataset with the proper classification for retraining and refining after being revised by a supervisor.

### **1.3.7 Notification**

The system is going to notify the user of new messages and also notify that he/she received a bully message and it has been blocked.

### **1.3.8 Training**

The Training module is the module where the system learns either from a data set or from the approved wrong classified messages from the learning module and this module is adds more samples to the classifier for better classification.

## **1.4 Business Context**

Moreover as to a business point of view more and more about cyberbullying will be dug through the vast extent of social media platforms as incidents of cyberbullying have doubled throughout the past 5 years.

1. Vision: Safe chatting for all people that are using social media.
2. Mission: Develop application that detect cyberbullying at the running time to help social media platforms and support the campaign that is happening

nowadays that want to end all bullying if not then reduce it

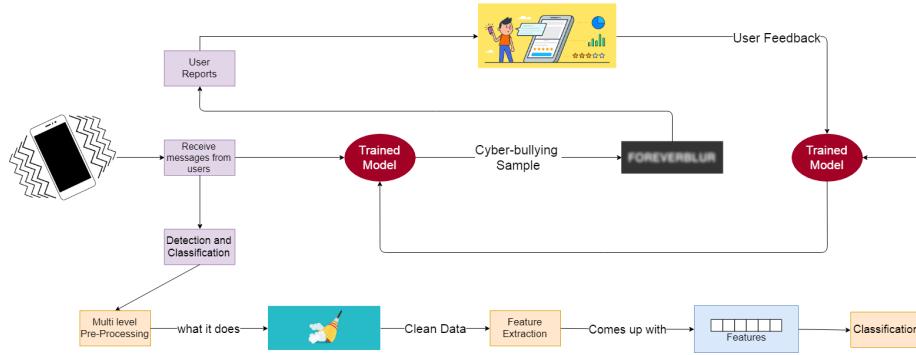


Figure 3: Business Model

## 2 General Description

### 2.1 Product Functions

Our bullying detection system consists of eight modules that are  
 1-Messages:It will receive this messages from Twitter or Facebook or Whats app to our application then it will send this message to the classifier.

2-Pre-processing: it will take the dataset or the message and it will cut them into sentence then it will make ,Stemming, Lemmatization, Remove Encoding and Word correction using Bing word correction.

3-Training: There will be Training to the three classifiers that are SVM, Naive Bayes and Random forest.

4-Notification: The system will notify the client if there is cyberbullying in the message that comes to him and the system will notify the admin if there is error in the system.

5-Learning: the system will make the user rate the accuracy of its detection then this ratings will go the modulator to be sure that is cyberbullying then to train the classifier with the new messages.

6-Report:It will appear to the user of the system daily to show him the events that happened in these day.

7-User management: this module will have interaction with the user and it will deal with user Login, register and changing of user type.

8-Classifier this module will take message from message module and give result is it cyberbullying or not by voting between three classifiers.

#### **User Functions**

1. sign in or 2. sign up as his role in the system as user
3. send message
4. Receive Message
5. void Rating(rate message)

As a customer he can update his own info.

- 6.Update

Admin will be able to see the errors

- 7.Message Errors

The developer can access the preprocessing and feature extraction and the processing

**File manager functions**

1. CSV
2. read Text

**Pre-processing module**

1. Read file.
2. Tokenization.
3. Stemming.
4. Lemmatization
5. Remove Encoding
6. Word correction

**Feature extraction**

1. TFIDF
2. Sentiment features

**Processing**

- (a) Fit()
  - (b) Prediction()
  - (c) Processing()
1. Training
  2. SVM
  3. Neural Network

The system will encrypt and decrypt the messages

**User functions**

The system will notify the user

**Notification functions**

1. Notify user
2. Register user
3. Remove user
4. Update Notification

All User information will held and controlled by our database

**Database Functions**

1. add
2. delete
3. modify

## 2.2 Similar System Information

[www.noswearing.com](http://www.noswearing.com) it is website that filters the sentences from curse or slang words and replace it with words have the same meaning of it .It is like the sites that change slang to normal language. They collect the bad words as the bag of words and they put it in a list so when the application find it .it will change it to another word with the same meaning.



Figure 4: Swear Word List and Curse Filter

### 2.3 User Characteristics

This project aims to target large number of people who use messenger applications. The users of our applications hope to not be bullied during their conversations on our application. Our application will detect the people who are making cyberbullying and report them. The user must be connected to the internet.

### 2.4 User Problem Statement

Data on the internet nowadays is too huge to be monitored manually by humans to detect cyberbullying and people use messenger applications multiple times per day so they might be exposed to bullying messages. In previous cyberbullying detection papers there has been a problem in detecting false positive cyberbullying's cases. The accuracy in these papers is not high enough and could be improved. Sarcasm which is type of cyberbullying is not detected in these papers.

## 2.5 User Objectives

The application is designed for individual users who are using messenger applications, like WhatsApp and Facebook Messenger to detect cyberbullying in messages. And the users can report these messages.

## 2.6 General Constraints

Our system constraints is the ambiguity of words (Contextual Analysis) also the high rate of false positive instances. It is used mainly for smart phones, which are capable of having internet connection as well as having our unique Chatting application .

# 3 Functional Requirements

## 3.1 Send \_ Message

FRID	1
Description	This function sends Messages from user to the platform that the user signed in on and the function takes the message and id of user as a parameter
Action	Sends a message to the social media platform
Input	String and Integer
Output	Boolean
Precondition	the user must be logged in
Post-condition	None
Dependencies	
Priority	10 \ 10

Table 1: Send \_ Message

### 3.2 Receive \_ Message

FRID	2
Description	This function Receive Messages from the platform API
Action	Checks for any message sent to the user and then return the received message
Input	None
Output	Message object
Precondition	the user must be logged in
Post-condition	None
Dependencies	
Priority	10 \ 10

Table 2: Receive \_ Message

### 3.3 Preprocessing

FRID	3
Description	This function do some preprocessing on the received messages like: Stemming, Lemmatization, Tokenization, RemoveEncoding and WordCorrection
Action	Takes message as String and apply all of the preprocessing tools
Input	String
Output	one dimensional List
Precondition	There is text to process
Post-condition	The preprocessed text then goes to the feature extraction
Dependencies	
Priority	10 \ 10

Table 3: Preprocessing

## 3.4 Extract \_ Features

### 3.4.1 TFIDF

FRID	4
Description	This function extract features from the preprocessed text using TFIDF function
Action	Takes preprocessed text and extract features from it
Input	one dimensional List
Output	List of 2d numpy array
Precondition	There is preprocessed text to extract the feature from
Post-condition	Then goes to the classifier to classify the object
Dependencies	
Priority	10 \ 10

Table 4: TFIDF

### 3.4.2 LIWC

FRID	5
Description	This function extract features from the preprocessed text using LIWC function
Action	Takes preprocessed text and extract features from it
Input	[]List
Output	[]List
Precondition	There is preprocessed text to extract the feature from
Post-condition	Then goes to the classifier to classify the object
Dependencies	
Priority	10 \ 10

Table 5: LIWC

### 3.4.3 Sentiment\_Analysis

FRID	6
Description	This function extract features from the preprocessed text using Sentiment Analysis
Action	Takes preprocessed text and extract features from it
Input	[[List
Output	]]List
Precondition	There is preprocessed text to extract the feature from
Post-condition	Then goes to the classifier to classify the object
Dependencies	
Priority	10 \10

Table 6: Sentiment\_Analysis

## 3.5 Classification

### 3.5.1 Classify

FRID	7
Description	This function takes the extracted features of the text, train the classifier and predict the class of the sent message
Action	take the extracted features to train the classifier and classify the sent message
Input	two dimensional list for the text and one dimensional list for the classes
Output	None
Precondition	There is a sent message that needs to be classified
Post-condition	classify data
Dependencies	
Priority	10 \10

Table 7: Classification

### 3.6 SignUp

FRID	8
Description	This function creates account for users
Action	Takes Data of user and insert in database
Input	String firstName, String LastName,int age, String gender, String email
Output	Boolean
Precondition	That the user doesn't exists
Post-condition	Account created
Dependencies	
Priority	10 \ 10

Table 8: SignUp

### 3.7 Update \_ Classifier

FRID	9
Description	This function takes the extracted features of new data and updates the classifier with it
Action	Train the classifier with new data
Input	[]list
Output	None
Precondition	There is new data to train the classifier with
Post-condition	Classify data
Dependencies	
Priority	10 \ 10

Table 9: Update \_ Classifier

### 3.8 Login

FRID	10
Description	This function verifies user's account
Action	Check for username and password in database
Input	String user-name & String password
Output	Boolean
Precondition	User has an account already
Post-condition	None
Dependencies	The application can't be accessed without being logged in
Priority	10 \ 10

Table 10: Login

### 3.9 Send \_ Notification

FRID	11
Description	This function sends notifications
Action	Sends notification from cloud to application
Input	String
Output	Boolean
Precondition	None
Post-condition	None
Dependencies	
Priority	10 \ 10

Table 11: Send \_ Notification

### 3.10 Encrypt

FRID	12
Description	This function encrypt messages
Action	Encrypt messages before being sent and encrypt user data before storing it in the database
Input	String
Output	String
Precondition	Data needed to be encrypted
Post-condition	Encrypted data
Dependencies	
Priority	9 \ 10

Table 12: Encrypt

### 3.11 Decrypt

FRID	13
Description	This function decrypt Data and messages
Action	Decrypt messages when the messages are received
Input	String
Output	String
Precondition	Data needed to be decrypted
Post-condition	Decrypted data
Dependencies	
Priority	9 \10

Table 13: Decrepit

### 3.12 Delete \_ Message

FRID	14
Description	This function deletes messages that the user selected
Action	Delete message from database and UI
Input	None
Output	Boolean
Precondition	The id exists in database
Post-condition	The message has been deleted
Dependencies	
Priority	8 \10

Table 14: Delete \_ Message

### 3.13 Copy \_ Message

FRID	15
Description	This function takes the selected messages and put them in clipboard for further operations
Action	Copies message to clipboard
Input	Message object
Output	Boolean
Precondition	THere is message selected
Post-condition	The message has been copied to the clipboard
Dependencies	
Priority	8 \10

Table 15: Copy \_ Message

### **3.14 Paste \_ Message**

FRID	16
Description	This function paste messages
Action	print message from clipboard on screen
Input	Message object
Output	Boolean
Precondition	The id exists in clipboard
Post-condition	The message is pasted on screen
Dependencies	
Priority	8 \10

Table 16: Paste \_ Message

### **3.15 Logout**

FRID	17
Description	This function logs the user out of the application
Action	Logout and redirect to login screen
Input	None
Output	Boolean
Precondition	The user must be logged in
Post-condition	Go to login screen
Dependencies	For the user to use the application he must login again
Priority	7 \10

Table 17: Logout

### **3.16 Show \_ Notification**

FRID	18
Description	This function is for notification displaying
Action	Shows notification on screen
Input	String
Output	Boolean
Precondition	None
Post-condition	The notification is printed in screen
Dependencies	
Priority	7 \10

Table 18: Show \_ Notification

## 4 Interface Requirements

### 4.1 User Interfaces

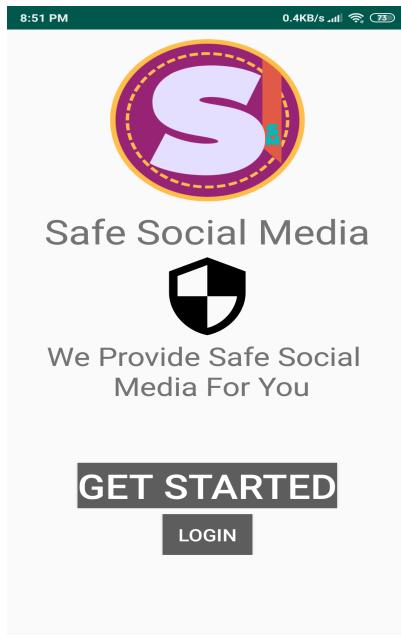


Figure 5: Rating

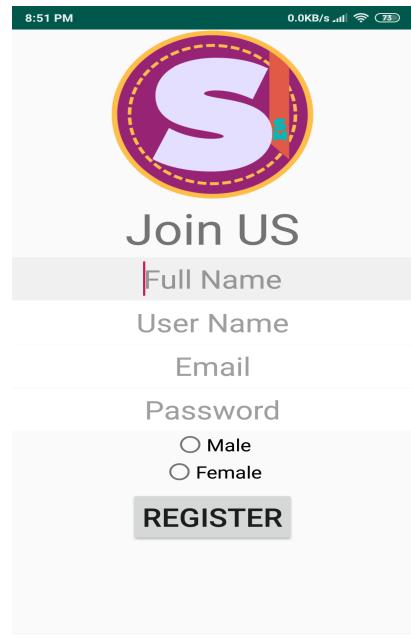


Figure 6: Register

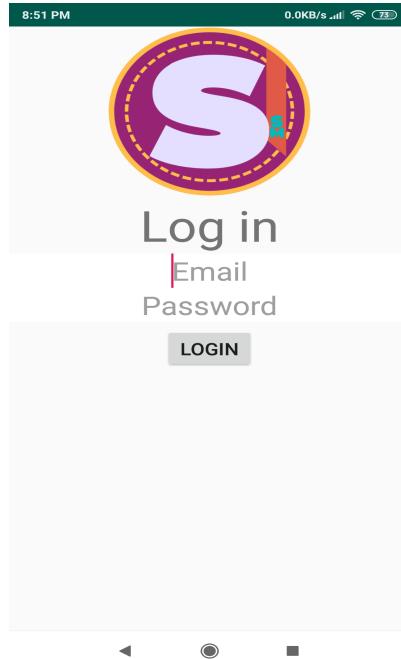


Figure 7: Log in

#### 4.1.1 API

1. Bing word Correction API
2. Microsoft Azure VM API

## 5 Performance Requirements

The application will be fast as we know that the application will send the messages to the cloud to detect if there is cyberbullying in the message or not. This process must never consume a lot of time and we aim to make it in less than one second. Our application will use memory as the other freeware and cross-platform messaging service we estimate that it will use 100 MB as average usage.

## 6 Design Constraints

### 6.1 Standards Compliance

Our application will need an android device that is connected to the internet that we will install the application on it. And the application will be installed on the cloud.

## **6.2 Hardware Limitations**

The phone must be at least 512 Ram also it must contain appropriate storage space to put the application. and it must be a smart phone.

## **6.3 others as appropriate**

# **7 Other non-functional attributes**

## **7.1 Security**

The data between the user app and the host(Trained Model) Should be encrypted because it consists of Private Messages and also the login credentials for the user in our system should be highly encrypted.

## **7.2 Portability**

The Software is going to be portable as it is going to be available for mobile environments and could make it support any platform as the core of the application is on a cloud and this application is just a client that sends the data for the core to process it.

## **7.3 Maintainability**

Our Proposed Application should be maintainable as it learns its self from the previous wrong classifications by the help of the user feedback on miss-classified instances and also by the reporting of the user to the wrong classifications.

# 8 Preliminary Object-Oriented Domain Analysis

## 8.1 Inheritance Relationships

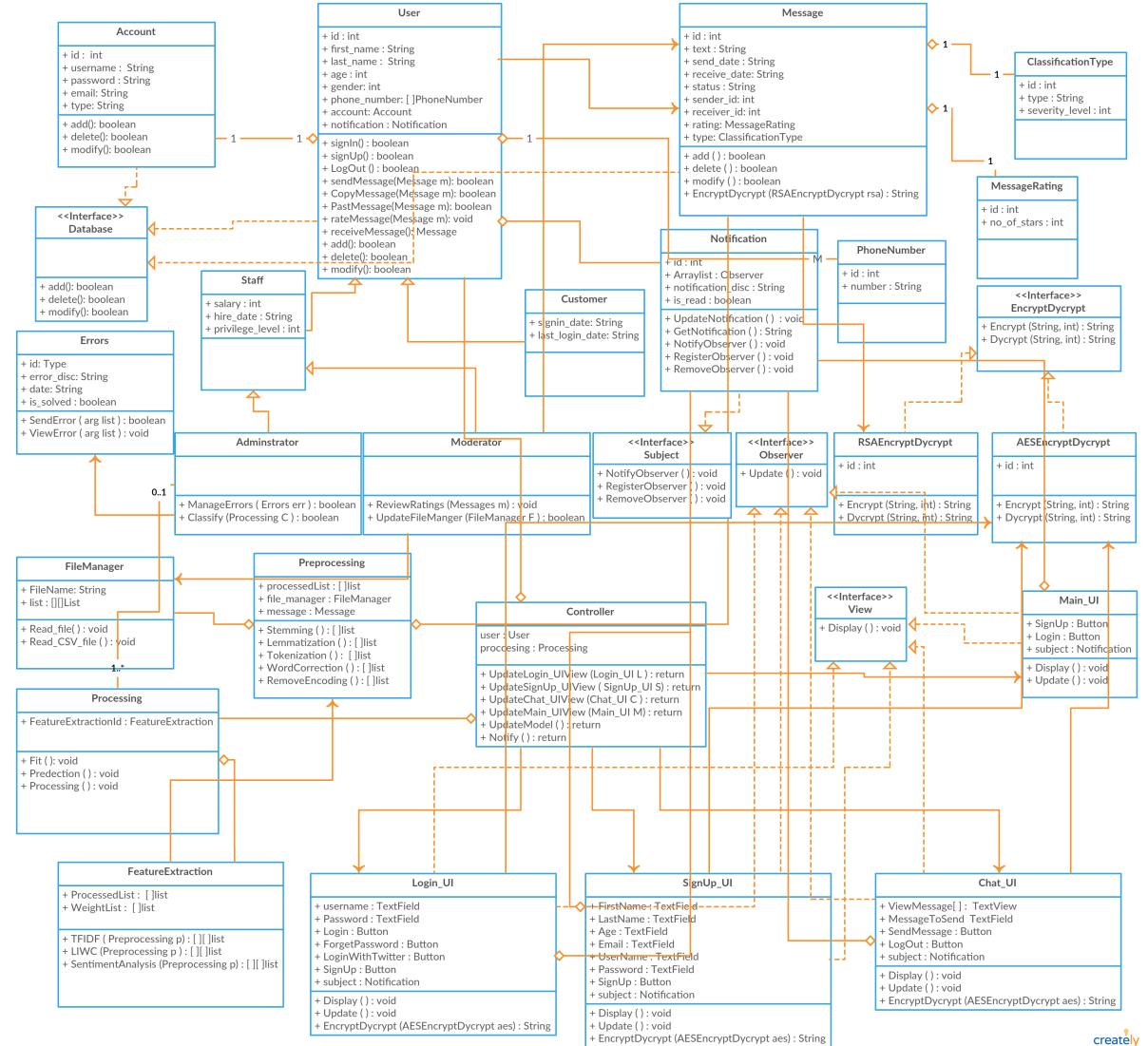


Figure 8: Class Diagram

## 8.2 Class descriptions

### 8.2.1 User Class

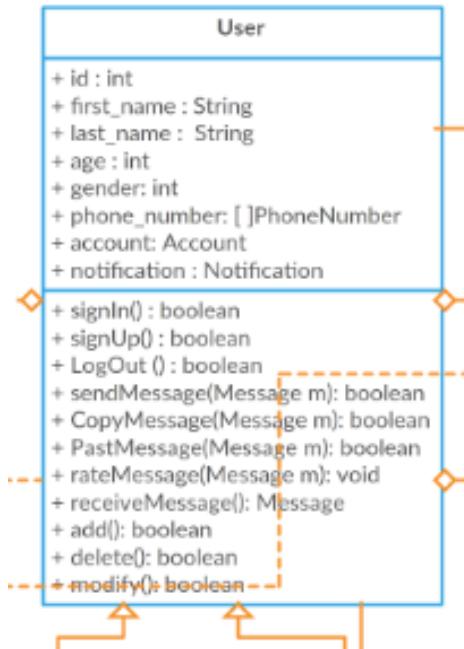


Figure 9: User class

Class name: User

List of super classes: N/A

List of sub classes: Administrator and customer.

Purpose: this class is used hold user info inside it

Collaborations: this class has account for security and has message as object in this class.

Attributes :

- (a) int :id
- (b) string : firstName
- (c) string : lastName
- (d) string : email
- (e) account Account
- (f) Int : Gender
- (g) Int : Age
- (h) Int : Gender
- (i) Int : Age
- (j) phonenumbers[] : phonenumbers

Operations :

- (a) boolean sign in( ) : creates a session for the user to stay logged in through all pages
- (c) void signUP ( ) : creates new user in the database
- (d) boolean send message( ) : Send message to another user
- (d) boolean Copy message( ) : Copy message to clipboard
- (d) boolean Past message( ) : Past message from clipboard to UI
- (e) Message Recicve Message( ) : Recive message from another user
- (f) void Rating( ) : Rate the prectetage of bulleying in mesaage from 1 to 5.

### 8.2.2 Customer Class

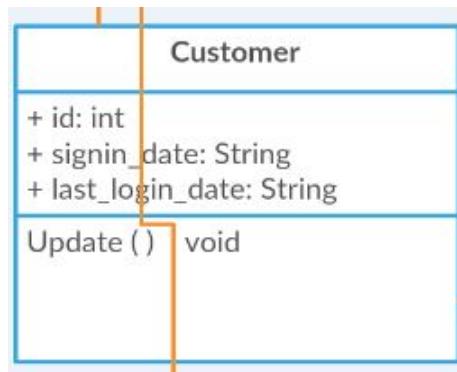


Figure 10: Customer class

Class name: Customer

List of super Classes :

(a) User

List of sub Classes : N/A

Purpose: this class is used hold Customer Date inside it

Attributes

(b) string: Sign in date

(C) string Last sign Up date

Operations :

### 8.2.3 Administrator Class

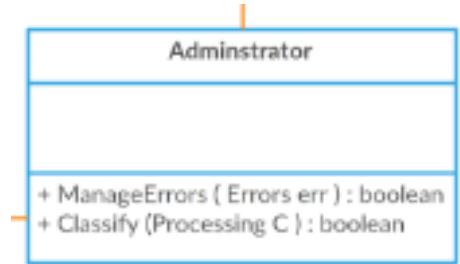


Figure 11: Adminestretor class

Class name: Administrator

List of super Classes :

(a) User

List of sub Classes : N/A

Purpose: this class is used to do some function for the administrator

Collaborations:

Attributes

Operations :

(a) Boolean ManageErrors(Errors err)

(b) Boolean Classify(Processing C)

#### 8.2.4 Error Class

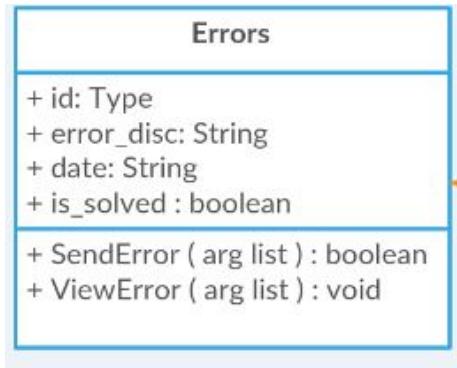


Figure 12: Error class

Class name: error

List of super Classes : N/A

List of sub Classes : N/A

Purpose: this class is used for viewing errors and sending them.

Collaborations: this class gives functions to the customer Class.

Attributes

- (a) int : id
- (b) string : Error Disc
- (c) String: Date
- (d) Boolean : is solved

Operations :

- (a) Boolean sendError( ) : Send Error to the admin
- (b) Void View Error() : View Error to user.

### 8.2.5 Notification Class

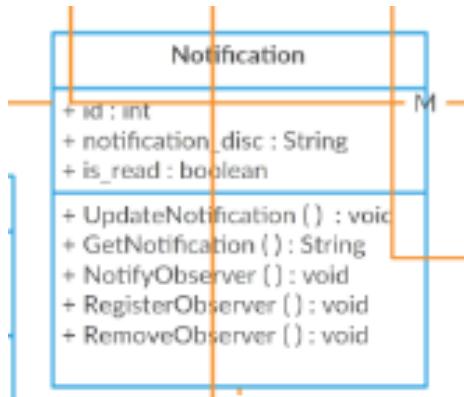


Figure 13: Notification class

Class name: notification

List of super Classes : N/A

List of sub Classes: N/A

Purpose: Sending notifications to the users.

Collaborations: this class will be aggregated from class user

Attributes

- (a) int : id
- (b) ArrayList : observers
- (c) string : Notification Disc
- (d) Boolean : is Read

Operations

- (a)Notify observer( )
- (b)Register observer()
- (c)Remove observer()
- (d)UpdateNotification()
- (e)GetNotification()

### 8.2.6 Phone Number Class

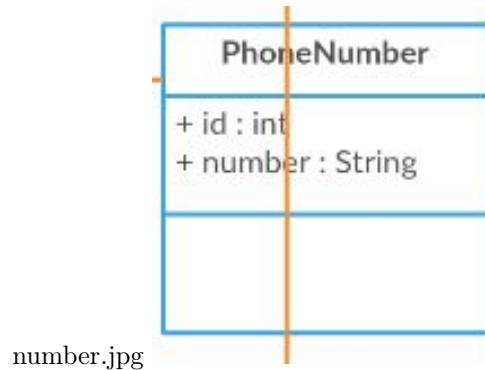


Figure 14: Phone Number

Class name : PhoneNumber

List of Superclasses : User

List of Subclasses : N/A

Purpose : hold the numbers of all users

Collaborations : Inheritance from class User , Association with class Database

Attributes :

- (a) int:id
- (b) String:number
- (c) int:UserId

Operations : N/A

Constraints : UserId is inherited from the User class properly as well as having the number correctly entered to be stored with no issues in the database

### 8.2.7 Processing class

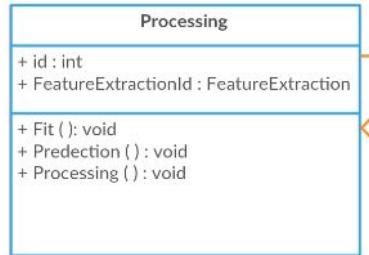


Figure 15: Processing class

Class name : Processing

List of Superclasses : N/A

List of Subclasses :

(a) FeatureExtraction

Purpose : First it holds the data being pushed from the FeatureExtraction class and then it's being classified as bullying or not

Collaborations :

(a) Aggregation from class FeatureExtraction

Attributes :

(a) int : id

(b) FeatureExtraction : FeatureExtractionId

Operations :

(a) Fit()

(b) Prediction()

(c) Processing()

Constraints : the array being pushed to the Processing class needs to be in correct values otherwise incorrect classification may occur.

### 8.2.8 File Manger Class

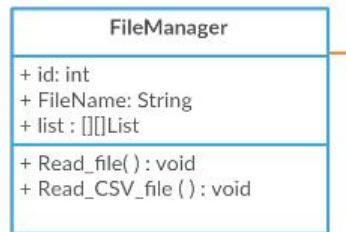


Figure 16: File Manager class

Class name : FileManager

List of Superclasses :

(a) Preprocessing

List of Subclasses : N/A

Purpose : deals with the dataset in order for it to be ready to be pushed to the Preprocessing class

Collaborations :

(a) Aggregates to Preprocessing

Attributes :

(a) int:id

(b) String:FileName

Operations :

(a) CS()

(b) readText()

Constraints : The dataset has to be modified correctly to be transferred correctly to the preprocessing class

### 8.2.9 Preprocessing Class

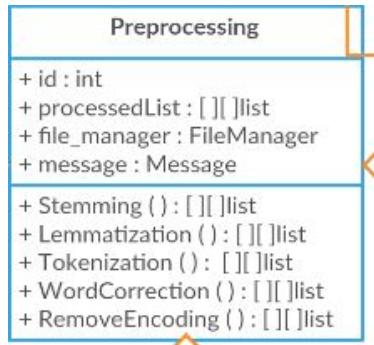


Figure 17: Preprocessing class

Class name : Preproceesing

List of Superclasses : N/A

List of Subclasses :

- (a) FileManager
- (b) FeatureExtraction

Purpose : the raw data being entered is being formatted and correctly analyzed and being prepared for feature extraction

Collaborations :

- (a) Aggregation from FileManager
- (b) Inheritance from FeatureExtraction

Attributes :

- (a) int : id
- (b) List : ProcessedList
- (c) int : FileManagerId

Operations :

- (a) Stemming()
- (b) Lemmatization()
- (c) Tokenization()
- (d) WordCorrection()
- (e) RemoveEncoding()

Constraints : data has to be passed on to the class with the correct formatting otherwise it would be useless from the beginning

### 8.2.10 Feature Extraction Class

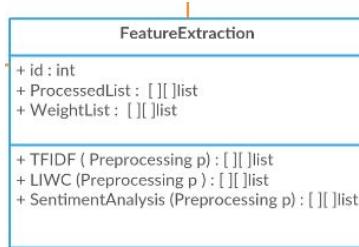


Figure 18: Feature Extraction class

Class name : FeaturExtraction

List of Superclasses :

(a) Preprocessing

List of Subclasses :

(a) Processing

Purpose : now that the data has been correctly formatted this class mostly deals with gathering information from the dull text and make it enriched with valuable meaning to be later passed on to the Processing phase

Collaborations :

(a) Inheritance from Preprocessing

(b) Aggregates to Processing

Attributes :

(a) int : id

(b) List : ProcessedList

(c) List : WeightList

Operations :

(a) TFIDF()

(b) LIWC()

(c) SentimentAnalysis()

Constraints : lists being passed from the preprocessing must be filled correctly and passed on with the correct formatting otherwise it would be difficult to extract meaningful values from the raw text being entered

### 8.2.11 Account Class

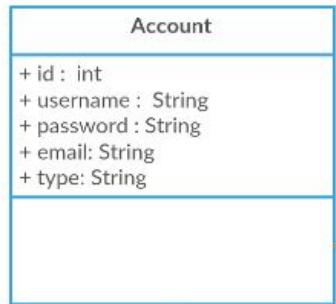


Figure 19: Account class

Class name : Account

List of Superclasses :

(a) User (b) Database List of Subclasses : N/A

Purpose : It holds the account details of every user.

Collaborations :

- (a) Aggregation to class User
- (b) Inheritance from class Database

Attributes :

- (a) int : id
- (b) String : username
- (c) String : password
- (d) String : email
- (e) String : type

Operations : N/A

Constraints : nn

### 8.2.12 Message Class

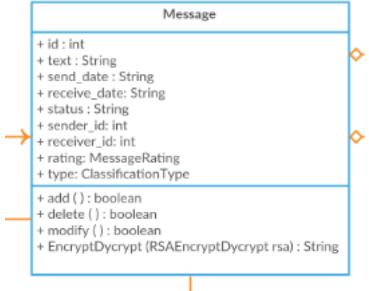


Figure 20: Message class

Class name : Message

List of Superclasses :

(a) Preprocessing

List of Subclasses :

(a) ClassificationType

(b) MessageRating

Collaborations :

(a) Aggregation from class Classification type

(b) Aggregation from class MessageRating

(c) Aggregation to class Preprocessing

Attributes :

(a) int : id

(b) String : text

(c) String : send-date

(d) String : recieve-data

(e) String : status

(f) int : sender-id

(g) int : reciever-id

(h) MessageRating : rating

(i) ClassificationType : type

Operations :

(a) boolean add ()

(b) boolean delete ()

(c) boolean modify ()

(d) String EncryptDcrypt (RSAEncryptDcrypt rsa)

Constraints : nn

### 8.2.13 Message Rating Class



rating.jpg

Figure 21: Message Rating class

Class name : MessageRating

List of Superclasses :

(a) Message

List of Subclasses : N/A

Collaborations :

(a) Aggregation to class Message

Attributes :

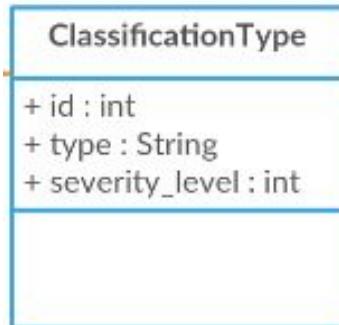
(a) int : id

(b) int : no-of-stars

Operations : N/A

Constraints : nn

#### 8.2.14 Classification Type Class



type.jpg

Figure 22: Classification Type class

Class name : ClassificationType

List of Superclasses :

(a) Message

List of Subclasses : N/A

Collaborations :

(a) Aggregation to class Message

Attributes :

(a) int : id

(b) String : type

(c) int : severity-level

Operations : N/A

Constraints : nn

### 8.2.15 Login\_UI Class

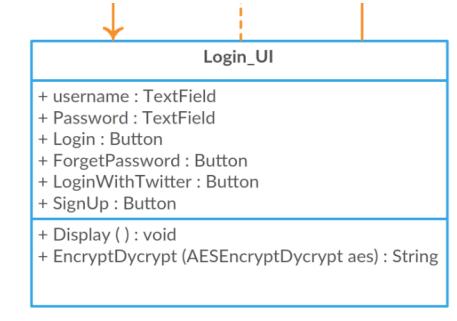


Figure 23: Login UI

1-Class name : Login\_UI

2-List of Superclasses : View, EncrypteDycrypt

3-List of Subclasses : None

4-Purpose : This class is used to login to your account

5-Collaborations : This class implements "View" interface to be able to draw all the views on the UI and it implements "EncrypteDycrypt" to encrypte and dycrypte Login information

6-Attributes :

(a)TextField: username

(b)TextField: Password

(c)Button: Login

(d)Button: ForgetPassword

(e)Button: LoginWithTwitter

(f)Button: SignUp

(g)Notification: subject

7-Operations :

(a) String EncryptDycrypt (AESEncryptDycrypt aes)

(b) void Display()

(C) void Update()

### 8.2.16 SignUp\_UI Class

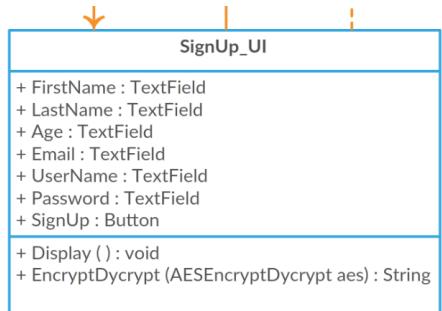


Figure 24: SignUp UI

1-Class name : SignUp\_UI

2-List of Superclasses : View, EncrypteDycrypte

3-List of Subclasses : None

4-Purpose : This class is used to Create your new account

5-Collaborations : This class implements "View" interface to be able to draw all the views on the UI and it implements "EncrypteDycrypte" to encrypte and dycrypte signUp information

6-Attributes :

- (a) TextField: FirstName
- (b) TextField: LastName
- (c) TextField: Age
- (d) TextField: Email
- (e) TextField: UserName
- (f) TextField: Password
- (g) Button: SignUp
- (h) Notification: subject

7-Operations :

- (a) String EncryptDycrypt (AESEncryptDycrypt aes)
- (b) void Display()
- (C) void Update()

### 8.2.17 Chat\_UI Class

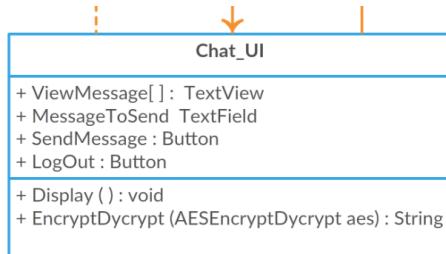


Figure 25: Chat UI

1-Class name : Chat\_UI

2-List of Superclasses : View, EncrypteDycrypte

3-List of Subclasses : None

4-Purpose : This class have the chat messages between two users

5-Collaborations : This class implements "View" interface to be able to draw all the views on the UI and it implements "EncrypteDycrypte" to encrypte and dycrypte all messages send and received

6-Attributes :

- (a) TextView: ViewMessage[]
- (b) TextField: MessageToSend
- (c) Button: SendMessage
- (d) Button: LogOut
- (e) Notification: subject

7-Operations :

- (a) String EncryptDycrypt (AESEncryptDycrypt aes)
- (b) void Display()
- (C) void Update()

### 8.2.18 Main\_UI Class

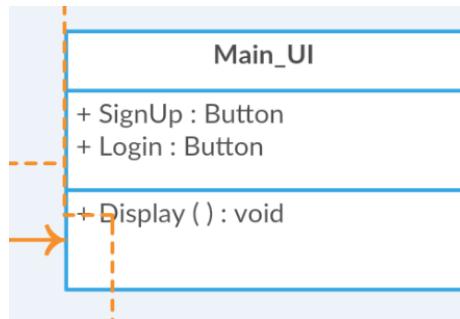


Figure 26: Main UI

- 1-Class name : Main\_UI
- 2-List of Superclasses : View
- 3-List of Subclasses : None
- 4-Purpose : This class is where the user choose either to login or signup
- 5-Collaborations : This class implements "View" interface to be able to draw all the views on the UI
- 6-Attributes :
  - (a)Button: SignUp
  - (b)Button: Login
  - (c)Notification: subject
- 7-Operations :
  - (a) void Display()
  - (b) void Update()

### 8.2.19 Controller Class

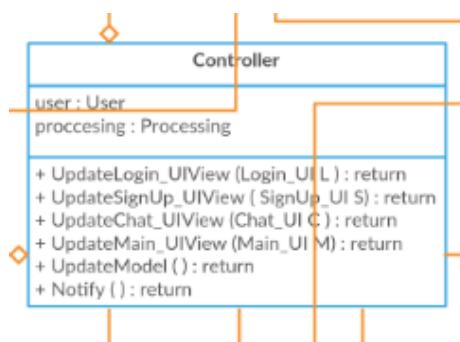


Figure 27: Controller

1-Class name : Controller  
 2-List of Superclasses : None  
 3-List of Subclasses : None  
 4-Purpose : This Class regulates the communication between models and views  
 5-Collaborations :  
 6-Attributes :  
 (a)User: user  
 (a)Processing: processing  
 7-Operations :  
 (a)void UpdateLogin\_UI(Login\_UI L)  
 (b)void UpdateSignUp\_UI(SignUp\_UI S)  
 (c)void UpdateChat\_UI(Chat\_UI C)  
 (d)void UpdateMain\_UI(Main\_UI M)  
 (e)void UpdateModel()

#### 8.2.20 RSAEncryptDycrypt Class

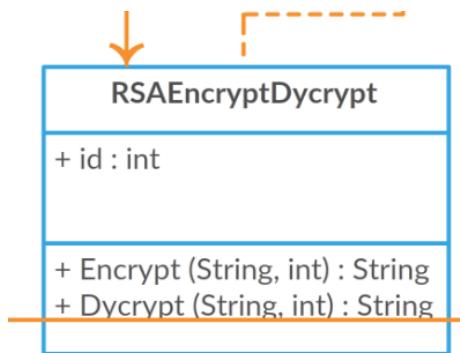


Figure 28: RSA

1-Class name : RSAEncryptDycrypt  
 2-List of Superclasses : EncryptDycrypt  
 3-List of Subclasses : None  
 4-Purpose :This class implements the RSA Encryption and Dycryption algorithm  
 5-Collaborations : The class implements "EncryptDycrypt" interface  
 6-Attributes :  
 (a)id : int  
 7-Operations :  
 (a)String Encrypt (String txt,int value)  
 (b)String Dycrypt (String txt,int value)

### 8.2.21 AESEncryptDycrypt Class

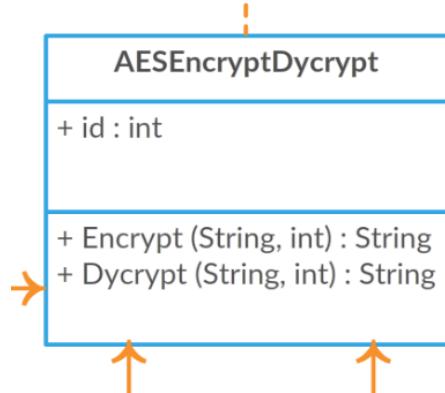


Figure 29: AES

- 1-Class name : AESEncryptDycrypt
- 2-List of Superclasses : EncryptDycrypt
- 3-List of Subclasses : None
- 4-Purpose : This class implements the AES Encryption and Dycryption algorithm
- 5-Collaborations : The class implements "EncryptDycrypt" interface
- 6-Attributes :
  - (a)id : int
- 7-Operations :
  - (a)String Encrypt (String txt,int value)
  - (b)String Dycrypt (String txt,int value)

### 8.2.22 Staff Class

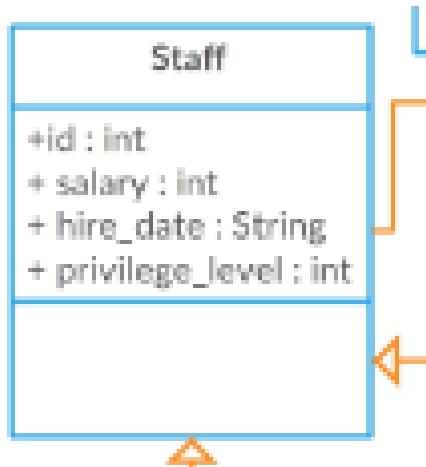


Figure 30: Staff

- 1-Class name : Staff
- 2-List of Superclasses : User
- 3-List of Subclasses : Administrator,Moderator
- 4-Purpose : This class holds the common information of working staff
- 5-Collaborations : The class extends "User" class
- 6-Attributes :
  - (a)salary : int
  - (b)hire\_date : String
  - (c)privilege\_level : int
- 7-Operations :

### 8.2.23 Moderator Class

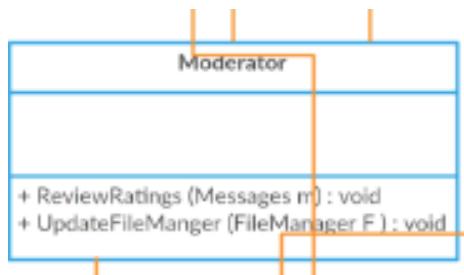


Figure 31: Moderator

- 1-Class name : Moderator
- 2-List of Superclasses : Staff,User
- 3-List of Subclasses :
- 4-Purpose : This class has some functions that Moderators use
- 5-Collaborations : The class extends "Staff" class
- 6-Attributes :
- 7-Operations :
  - (a) void ReviewRatings(Messages m)
  - (b) Boolean UpdateFileManger(FileManger F)

## 9 Operational Scenarios

### 9.1 Use Case

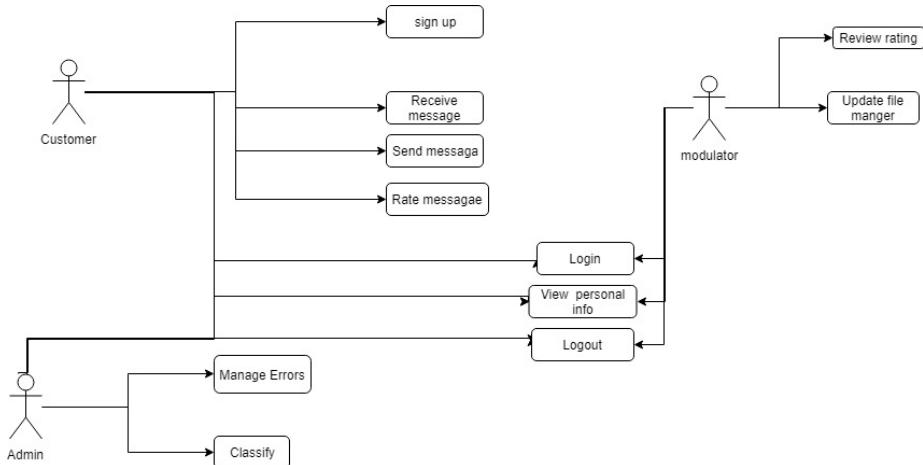


Figure 32: AES

Our operational scenario is as follows, we have a ready made dataset imported from kaggle website containing messages and their their weights, the dataset does appear to be labeled and supervised by the people who made it, first step is importing the dataset properly into our system and reading each message and value correctly in order for it to enter the preprocessing successfully, the dataset is now labeled and divided correctly in order for it to be entered to the feature extraction, in this phase meaningful values and expressions are extracted from each sentence and then passed on to the classifiers in order to come up with their classification whether it's cyber-bullying or not. Moving on to what will be implemented in the system. Which is having the user himself rate the messages that we have discovered as cyberbullying and then send that feedback to us for re-visioning, if its a match then we are doing something right, if not,

then that message will be sent back once again from the developers back to the training phase with a new classification for it and adapting the results to meet the users intended results. Yes there may be some issues here and there when it comes to user miss usage but it's a part we are willing to handle with care.

## 10 Preliminary Schedule Adjusted

Task Name	Start Time	Finish
Idea Discussion	1/8/2018	1/8/2018
Idea Research	1/8/2018	13/9/2018
Proposal Writing	13/9/2018	16/9/2018
Implementing Prototype	16/9/2018	17/9/2018
Delivering Rehearsal	18/9/2018	18/9/2018
Delivering Proposal	18/9/2018	26/9/2018
Doing Survey	10/10/2018	20/10/2018
Implementing Demo	20/10/2018	25/10/2018
Writing SRS	25/10/2018	30/10/2018
Training Model	30/10/2018	25/11/2018
Preparing For External Examiner	25/11/2018	3/12/2018
Building Desktop App	3/12/2018	18/1/2019
Writing SDD	18/1/2019	1/2/2019
Building Android App	1/2/2019	1/4/2019
Preparing For Implementation Evaluation	1/4/2019	25/4/2019
Writing 8 Pages Paper	25/4/2019	28/4/2019
Testing and Debuging Project	28/4/2019	7/5/2019
Writing Final Thesis	10/5/2019	25/5/2019
Presenting Final Thesis	25/6/2019	25/6/2019

## 11 Preliminary Budget Adjusted

Google Machine Learning engine 0.7usd/h

## 12 Appendices

### 12.1 Definitions, Acronyms, Abbreviations

Software delivery lifecycle (SDLC)

### 12.2 Collected material

N/A

## **13 References**

### **References**