UPCYCLER

A SENTIMENT ANALYSIS TOOL

SYEDA NAZISH KAZMI

USAMA MUKHTAR

SYED MUSTUFAIN ABBAS

Submitted to the Faculty of Computing Software Engineering

National University of Sciences and Technology, Islamabad in partial fulfillment for the requirements of a B.E Degree in Computer Software Engineering

ABSTRACT

UPCYCLER A SENTIMENT ANALYSIS TOOL

Opinions are one of the basic entities in human conversations. It is in human nature to have feelings and sentiments about the people and the happenings around them. Opinions and beliefs of people around us affect us in determining what choices we make and what do we believe in. Humans generally ask for opinion of other people when they need to make an important decision

Many applications like Facebook, Twitter and Messenger such as Whatsapp use text Data Mining and NLP techniques to extract relevant information from user conversations such as Customer reviews, popular news and general opinion of the public on certain topics. They then sell this information to the relevant brands, companies etc. who want to make use of this information.

We designed an application tool that can be used to extract useful information like user sentiments, security hazards and Customer Intelligence from the casual user conversations. The general idea is to make the useless natural language conversations useful by picking out only the meaningful information. Once, the information has been extracted, it can be used for different sorts of surveys and analysis. We used text from social media, blogs and news articles for this purpose. We have determined salient features of users like their opinions , sentiments, current trends , their interests and topics which user is talking about most .

CERTIFICATE FOR CORRECTNESS AND APPROVAL

Certified that work contained in the thesis- upcycler a sentiment analysis tool carried out by Syeda Nazish Kazmi ,Usama Mukhtar and Syed Mustufain Abbas under supervision of Dr. Hammad Afzal for partial fulfilment of Degree of Bachelor of Software Engineering is correct and approved.

Approved by

Dr.Hammad Afzal

CSE DEPARTMENT

MCS,NUST

DATED:

_____

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

In the name of Allah, the Most Merciful, the Most Beneficent. To our parents, without whose unflinching support and unstinting cooperation, a work of this magnitude would not have been possible.

ACKNOWLEDGEMENTS

To begin with, there is no greater guide than **ALLAH (SWT)** Himself and we feel blessed that He gave us enough strength to complete this project well in time. In addition to this, we all would deeply and genuinely like to thank Sir **Hammad Afzal** for his persistent guidance and continuous support. Sir you are an exceptional supervisor and without you we could not have come this far.

Table of Contents

List of figures

1. Introduction

Up cycling is the process of putting useless things into use.UPCYCLER makes use of apparently ‘useless’ data obtained from text from social media, blogs and news articles by extracting useful information from them. Information extracted will be used for applications like Sentiment Analysis and Threat Detection.

1.1 Purpose

Opinions are one of the basic entities in human conversations. It is in human nature to have feelings and sentiments about the people and the happenings around them. Opinions and beliefs of people around us affect us in determining what choices we make and what do we believe in. Humans generally ask for opinion of other people when they need to make an important decision. Feelings, whether our own or of anyone else, help us in evaluating the world around us. Sometimes, these feelings are positive and refreshing while at other times, they can be negative and hostile. For example, a Pakistani cricket fan will have refreshing feelings when Pakistan wins a cricket match against India but he will be saddened if Pakistan loses the next match.

Human behavior has always been an interesting subject. We, the humans have always wanted to be able to perceive how human nature works. It is important to learn about the existing human behaviors by analyzing how a person feels about a certain event or an individual. The opinions and sentiments of humans are of great value because these emotions actually lay the basis for human behavior.

The purpose of this document is to specify the requirements for the project, “UPCYCLER”. “UPCYCLER” is a stand-alone, application that can help us learn more about sentiments of people about certain entities like events, ideas, products and individuals. The audience of this document includes the development team, potential users and evaluation team.

1.2 Scope

Nowadays the data is overflowing so there is need to make use of this data and extract relevant information from it as this huge amount of data cannot be stored. It has become increasingly popular and useful. More and more research has been carried out in this aspect which includes Natural Language Processing, Data Mining, Text Mining and Web Mining. The growing popularity of Data Analysis can be estimated from the fact that other than IT professionals, successful business vendors and marketing companies have shown interest in the subject. Data analysis systems have found their applications in almost every business and social domain.

The project aims to develop a tool of such kind based on live stream of user conversations from a chat application. The application will be designed to deduce the public opinion and general sentiments of users about the current popular trends, politicians, company products and others. It will also be used to extract relevant information from the text from social media, blogs and news articles and would help to understand the significance of data by placing it in a visual context.

Opinions, sentiments, evaluations, attitudes, and emotions are all part of sentiment analysis and opinion mining. Popular sentiment analysis techniques and natural language processing algorithms will be extensively used in the development of product.

2. LITERATURE REVIEW

2.1 Introduction

Many applications like Facebook, Twitter and Messenger such as WhatsApp use text Data Mining and NLP techniques to extract relevant information from user conversations such as Customer reviews, popular news and general opinion of the public on certain topics. They then sell this information to the relevant brands, companies etc. who want to make use of this information.

2.2 Limitations of Prior Art

Sentiment analysis of in the domain of micro-blogging is a relatively new research topic so there is still a lot of room for further research in this area. Decent amount of related prior work has been done on sentiment analysis of user reviews, documents, web blogs/articles and general phrase level sentiment analysis. These differ from twitter mainly because of the limit of 140 characters per tweet which forces the user to express opinion compressed in very short text. The best results reached in sentiment classification use supervised learning techniques such as Naive Bayes and Support Vector Machines, but the manual labelling required for the supervised approach is very expensive. Some work has been done on unsupervised and semi-supervised approaches, and there is a lot of room of improvement. Various researchers testing new features and classification techniques often just compare their results to base-line performance. There is a need of proper and formal comparisons between these results arrived through different features and classification techniques in order to select the best features and most efficient classification techniques for particular applications.

2.3 Related Work

The bag-of-words model is one of the most widely used feature model for almost all text classification tasks due to its simplicity coupled with good performance. The model

represents the text to be classified as a bag or collection of individual words with no link or dependence of one word with the other, i.e. it completely disregards grammar and order of words within the text. This model is also very popular in sentiment analysis and has been used by various researchers. The simplest way to incorporate this model in our classifier is by using unigrams as features. Generally speaking n-grams is a contiguous sequence of “n” words in our text, which is completely independent of any other words or grams in the text. So unigrams is just a collection of individual words in the text to be classified, and we assume that the probability of occurrence of one word will not be affected by the presence or absence of any other word in the text. This is a very simplifying assumption but it has been shown to provide rather good performance. One simple way to use unigrams as features is to assign them with a certain prior polarity, and take the average of the overall polarity of the text, where the overall polarity of the text could simply be calculated by summing the prior polarities of individual unigrams. Prior polarity of the word would be positive if the word is generally used as an indication of positivity, for example the word “sweet”; while it would be negative if the word is generally associated with negative connotations, for example “evil”. There can also be degrees of polarity in the model, which means how much indicative is that word for that particular class. A word like “awesome” would probably have strong subjective polarity along with positivity, while the word “decent” would although have positive prior polarity but probably with weak subjectivity. There are three ways of using prior polarity of words as features. The simpler un-supervised approach is to use publicly available online lexicons/dictionaries which map a word to its prior polarity. The Multi-Perspective-Question-Answering (MPQA) is an online resource with such a subjectivity lexicon which maps a total of 4,850 words according to whether they are “positive” or “negative” and whether they have “strong” or “weak” subjectivity. The SentiWordNet 3.0 is another such resource which gives probability of each word belonging to positive, negative and neutral classes . The second approach is to construct a custom prior polarity dictionary from our training data according to the occurrence of each word in each particular class. For example if a certain word is occurring more often in the positive labelled phrases in our training dataset (as compared to other classes) then we can calculate the probability of that word belonging to positive class to be higher than the probability of occurring in any other class. This approach has been shown to give better performance, since the prior polarity of words is more suited and fitted to a particular type of text and is not very general like in the former approach. However, the latter is a supervised approach because the training data has to be labelled in the appropriate classes before it is possible to calculate the relative occurrence of a word in each of the class. It has been noticed a decrease in performance by using the lexicon word features along with custom n-gram word features constructed from the training data, as opposed to when the n-grams were used alone. The third approach is a middle ground between the above two approaches. In

this approach we construct our own polarity lexicon but not necessarily from our training data, so we don't need to have labelled training data. One way of doing this a is to calculate the prior semantic orientation (polarity) of a word or phrase by calculating it's mutual information with the word "excellent" and subtracting the result with the mutual information of that word or phrase with the word "poor" . They used the number of result hit counts from online search engines of a relevant query to compute the mutual information. The final formula they used is as follows:

$\boldsymbol{Pol}(\boldsymbol{phrase})$ = $\boldsymbol{log2hits}(\boldsymbol{phraseNEAR}$ "excellent"). ("Poor") $\boldsymbol{hit}(\boldsymbol{phraseNEAR}$ "poor"). $\boldsymbol{hits}$("$\boldsymbol{excellent}$")

Where hits (phrase NEAR "excellent") means the number documents returned by the search engine in which the phrase (whose polarity is to be calculated) and word "excellent" are co-occurring. While hits ("excellent") means the number of documents retuned which contain the word "excellent". With this idea and used a seed of 120 positive words and 120 negative to perform the internet searches. So the overall semantic orientation of the word under consideration can be found by calculating the closeness of that word with each one of the seed words and taking and average of it. Another graphical way of calculating polarity of adjectives has been discussed. . The process involves first identifying all conjunctions of adjectives from the corpus and using a supervised algorithm to mark every pair of adjectives as belonging to the same semantic orientation or different. A graph is constructed in which the nodes are the adjectives and links indicate same or different semantic orientation. Finally a clustering algorithm is applied which divides the graph into two subsets such that nodes within a subset mainly contain links of same orientation and links between the two subsets mainly contain links of different orientation. One of the subsets would contain positive adjectives and the other would contain negative. Many of the researchers in this field have used already constructed publicly available lexicons of sentiment bearing words (while many others have also explored building their own prior polarity lexicons The basic problem with the approach of prior polarity approach has been identified by Wilson et al. who distinguish between prior polarity and contextual polarity They say that the prior polarity of a word may in fact be different from the way the word has been used in the particular context. The paper presented the following phrase as an example: Philip Clapp, president of the National Environment Trust, sums up well the general thrust of the reaction of environmental movements: "There is no reason at all to believe that the polluters are suddenly going to become reasonable." In this example all of the four underlined words "trust", "well", "reason" and "reasonable" have positive polarities when observed without context to the phrase, but here they are not being used to express a positive sentiment. This concludes that even though generally speaking a word like "trust" may be used in positive sentences, but this doesn't rule out the chances of it appearing in non-positive sentences as well. Henceforth prior polarities of individual words (whether the words generally carry positive or negative connotations) may alone

not enough for the problem. The paper explores some other features which include grammar and syntactical relationships between words to make their classifier better at judging the contextual polarity of the phrase. The task of twitter sentiment analysis can be most closely related to phraselevel sentiment analysis. A seminal paper on phrase level sentiment analysis was presented in 2005 ,which identified a new approach to the problem by first classifying phrases according to subjectivity (polar) and objectivity (neutral) and then further classifying the subjective-classified phrases as either positive or negative. The paper noticed that many of the objective phrases used prior sentiment bearing words in them, which led to poor classification of especially objective phrases. It claims that if we use a simple classifier which assumes that the contextual polarity of the word is merely equal to its prior polarity gives a result of about 48%. The novel classification process proposed by this paper along with the list of ingenious features which include information about contextual polarity resulted in significant improvement in performance (in terms of accuracy) of the classification process. The results from this paper are presented in the table below:

Features	Accuracy	Subjective F.	Objective F.
Word tokens	73.6	55.7	81.2
Words + prior polarity	74.2	60.6	80.7
28 features	75.9	63.6	82.1

Table 2: Step 1 results for Objective / Subjective Classification in [16]

Features	Accuracy	Positive F.	Negative F.	Both F.	Objective F.
Word tokens	61.7	61.2	73.1	14.6	37.7
Word + prior	63.0	61.6	75.5	14.6	40.7
10 features	65.7	65.1	77.2	16.1	46.2

Table 3: Step 2 results for Polarity Classification in [16]

One way of alleviating the condition of independence and including partial context in our word models is to use bigrams and trigrams as well besides unigrams. Bigrams are collection of two contiguous words in a text, and similarly trigrams are collection of three

contiguous words. So we could calculate the prior polarity of the bigram / trigram - or the prior probability of that bigram / trigram belonging to a certain class – instead of prior polarity of individual words. Many researchers have experimented with them with the general conclusion that if we have to use one of them alone unigrams perform the best, while unigrams along with bigrams may give better results with certain classifiers .However trigrams usually result in poor performance as reported The reduction in performance by using trigrams is because there is a compromise between capturing more intricate patterns and word coverage as one goes to higher-numbered grams. Besides from this some researchers have tried to incorporate negation into the unigram word models. Researchers used a model in which the prior polarity of the word was reversed if there was a negation (like “not”, “no”, “don’t”, etc.) next to that word. In this way some contextual information is included in the word models. Project Thesis Report 19 Grammatical features (like “Parts of Speech Tagging” or POS tagging) are also commonly used in this domain. The concept is to tag each word of the tweet in terms of what part of speech it belongs to: noun, pronoun, verb, adjective, adverb, interjections, intensifiers etc. The concept is to detect patterns based on these POS and use them in the classification process. For example it has been reported that objective tweets contain more common nouns and third-person verbs than subjective tweets [3], so if a tweet to be classified has a proportionally large usage of common nouns and verbs in third person, that tweet would have a greater probability of being objective (according to this particular feature). Similarly subjective tweets contain more adverbs, adjectives and interjections [3]. These relationships are demonstrated in the figures below:

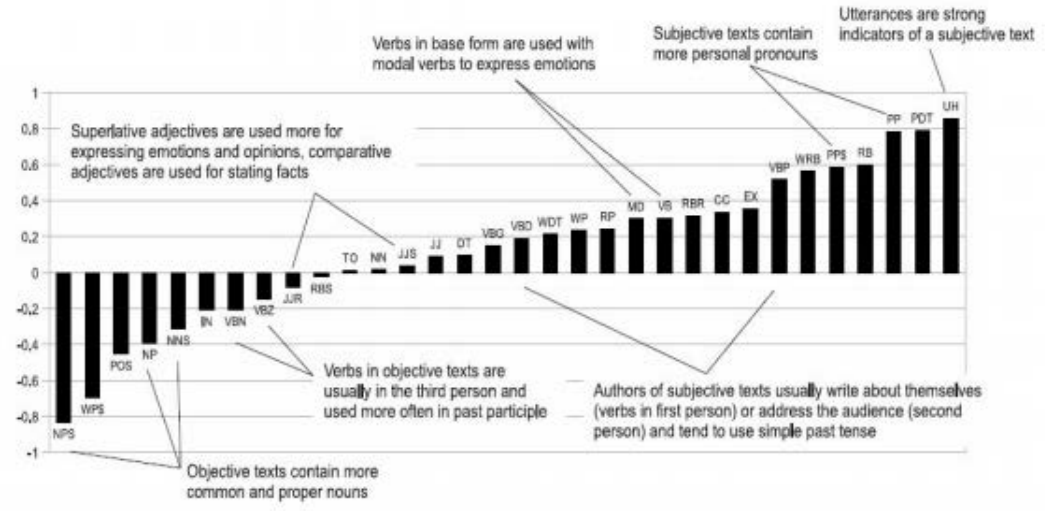

Figure 1: Using POS Tagging as features for objectivity/subjectivity classification

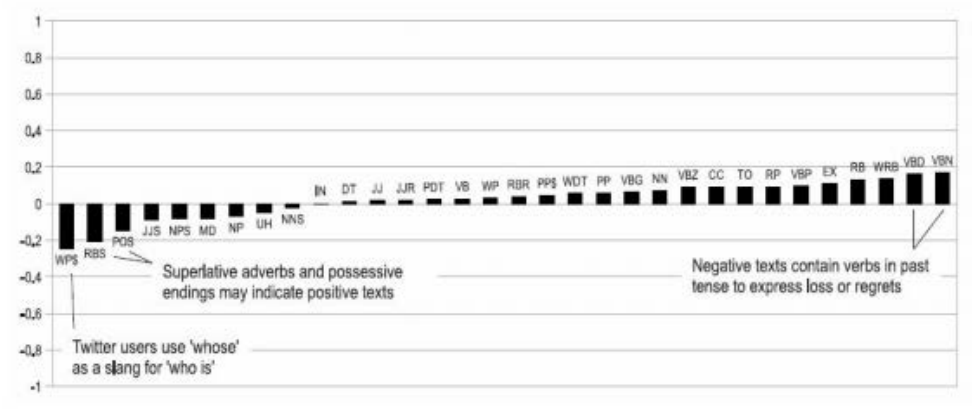

Figure 2: Using POS Tagging as features in positive/negative classification

However there is still conflict whether Parts-of-Speech are a useful feature for sentiment classification or not. Some researchers argue in favour of good POS features while others not recommending them .Besides from these much work has been done in exploring a class of features pertinent only to micro blogging domain. Presence of URL and number of capitalized words/alphabets in a tweet have been explored .Researchers have also reported positive results for using emoticons and internet slang words as features .Researchers does study on word lengthening as a sign of subjectivity in a tweet .The paper reports positive results for their study that the more number of cases a word has of lengthening, the more chance there of that word being a strong indication of subjectivity. The most commonly used classification techniques are the Naive Bayes Classifier and State Vector Machines. Some researcherspublish better results for SVMs while others support Naive Bayes and also report good results for Maximum Entropy classifier. Project Thesis Report 21 It has been observed that having a larger training sample pays off to a certain degree, after which the accuracy of the classifier stays almost constant even if we keep adding more labelled tweets in the training data used tweets labelled by internet resources instead of labelling them by hand, for training the classifier. Although there is loss of accuracy of the labelled samples in doing so (which is modelled as increase in noise) but it has been observed that if the accuracy of training labels is greater than 50%, the more the labels, the higher the accuracy of the resulting classifier. So in this way if there are an extremely large number of tweets, the fact that our labels are noisy and inaccurate can be compensated On the other hand researchers use presence of positive or negative emoticons to assign labels to the tweets .Like in the above case they used large number of tweets to reduce effect of noise in their training data. Some of the earliest work in this field classified text only as positive or negative, assuming that all the data provided is subjective .While this is a

good assumption for something like movie reviews but when analyzing tweets and blogs there is a lot of objective text we have to consider, so incorporating neutral class into the classification process is now becoming a norm. Some of the work which has included neutral class into their classification process there has also been very recent research of classifying tweets according to the mood expressed in them, which goes one step further. Researchers explores this area and develops a technique to classify tweets into six distinct moods: tension, depression, anger, vigor, fatigue and confusion .They use an extended version of Profile of Mood States (POMS): a widely accepted psychometric instrument. They generate a word dictionary and assign them weights corresponding to each of the six mood states, and then they represented each tweet as a vector corresponding to these six dimensions. However not much detail has been provided into how they built their customized lexicon and what technique did they use for classification.

3. SOFTWARE REQUIREMENT SPECIFICATION:-

3.1 Overview

The remaining sections of this document will cover an overall description of the project. The requirements and specifications for the software application are discussed in detail. The overall description of the software can be useful for the customers or users to determine whether the requirements specified fulfill their requirements and development team as well in the development of the product.

Here is an overview of the remaining section of this sections:

- **Overall Description** has the detailed description of the product in general. This section covers what the product is and how it can be used for practical applications.
- **External Interface Requirements**specifies the requirements for external interfacing for the application.
- **System Features** mainly specifies the functional requirements. The system functionality depends on this section.
- **Non Functional Requirements**covers the non-functional requirements like system reliability, portability etc.

3.2 Overall Description

This section has detailed description about the project. Basic features and constraints are discussed.

3.2.1. Product Perspective

This product is similar to Twitter Sentiment Analysis tool in which they fetch live tweets and extract sentiments and other relevant information from it. It is a web application with output in the form of a webpage.

3.2.2 Product Functions

The project “UPCYCLER” is similar to Twitter Sentiment Analysis Tool except it would use text from social media, blogs and news articles .Sentiments of users about certain topics would be extracted. It would also be used to extract other relevant information.

3.2.3 User Characteristics

The intended user will be a member of general public who is interested in the sentiments of the user. Public user can view trends in a timeline or enter a keyword to view a specific trend. They can also view user sentiments in text form as well as graphically to aid in better understanding of user sentiments about that trend. Authenticated users will have privileged access to the information that would not be visible to the public user. This information would contain user details Users are not expected to have a very high level of technical expertise.

3.2.4Constraints

User conversations would contain user conversation and the user id. Username will not be provided.

3.2.5 Assumptions and Dependencies

An assumption is that it is possible to accurately determine the sentiment for a 140 character string of English text.

3.3 External Interface Requirements

This section covers the interfacing requirements for our project. Features of user interfaces are described.

3.3.1 Hardware Requirements

The application is intended to be a stand-alone single user system. The application will be a web application. No further hardware devices or interfaces will be required.

3.3.2 Software Requirements

3.3.2.1 User Interfaces

The interface will meet the following requirements to conform to the users’ needs. It will be simple and easy to understand. Controls which allow the user to interact with the application will be clear and imply their functionality within the application. The interface

will include user inputs as well as graphical visualization of data. The graphs displayed to the user will provide a visual representation of the output. Error notifications will be required within the application, presenting the user with appropriate user conversations which describe the error that has taken place. If applicable, error user conversations should suggest possible solutions to the problem.

3.3.2.2 Input Interfaces

The user would be able to know the latest trends, past trends, what users are generally talking about or any topic by entering, removing or editing the keywords. The user would then be able to know the user sentiments about a certain topic.

3.3.2.3 Output Interfaces

The output to the user would be displayed graphically in a clear and meaningful manner which the user can easily understand.

3.4 System Features

System features are discussed in detail. This heading mainly covers the functional requirements of the project.

3.4.1 Retrieving Input (Discussion)

The software will receive two inputs: keyword sand text from social media, blogs and news articles.

- Keywords will be entered by the user for each topic.
- Text from social media, blogs and news articles will be retrieved with the (discussion).

3.4.2 Use Cases

AUTHENTICATED USER

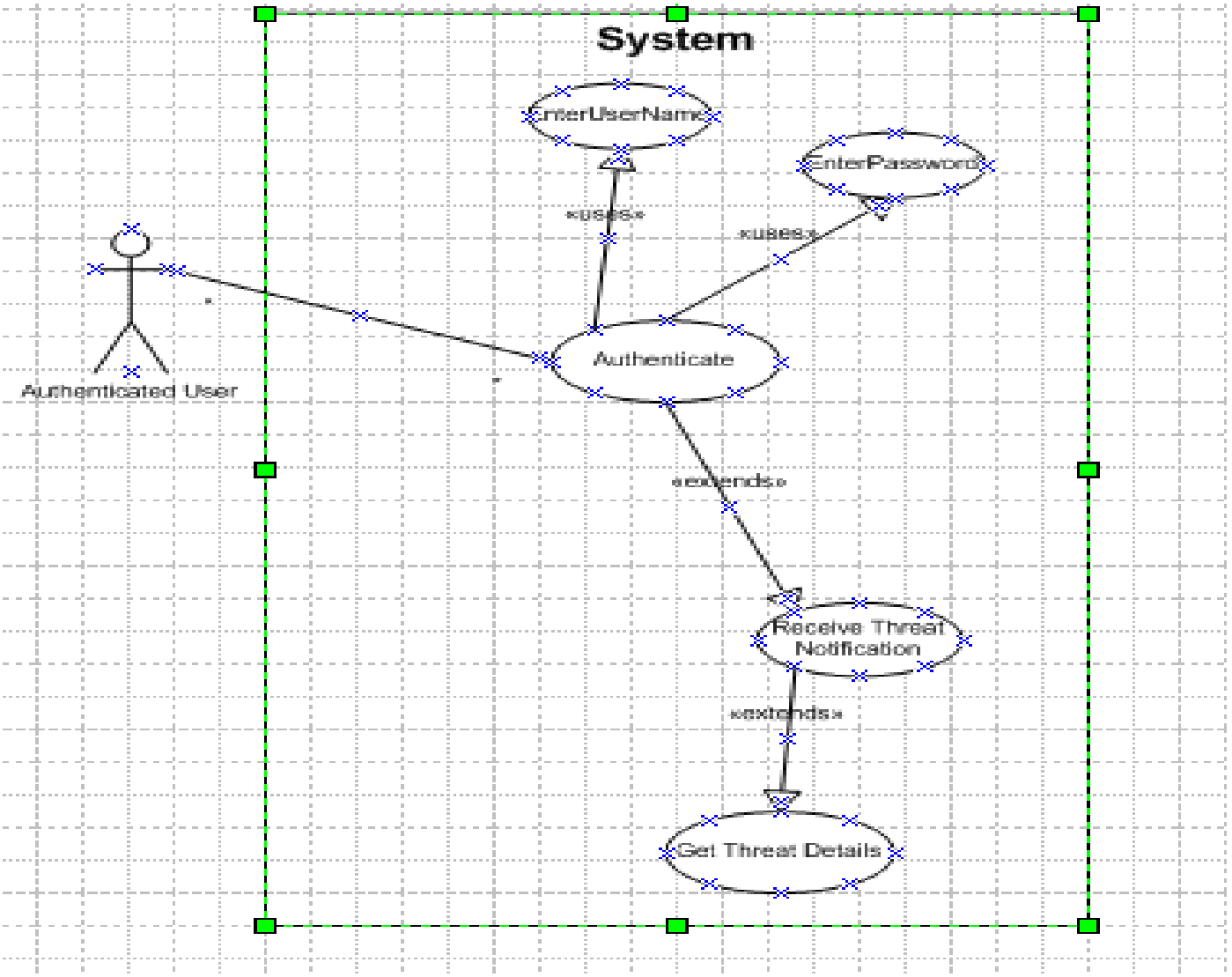

Figure 1-Use case Authenticated Use

GUEST USER:-

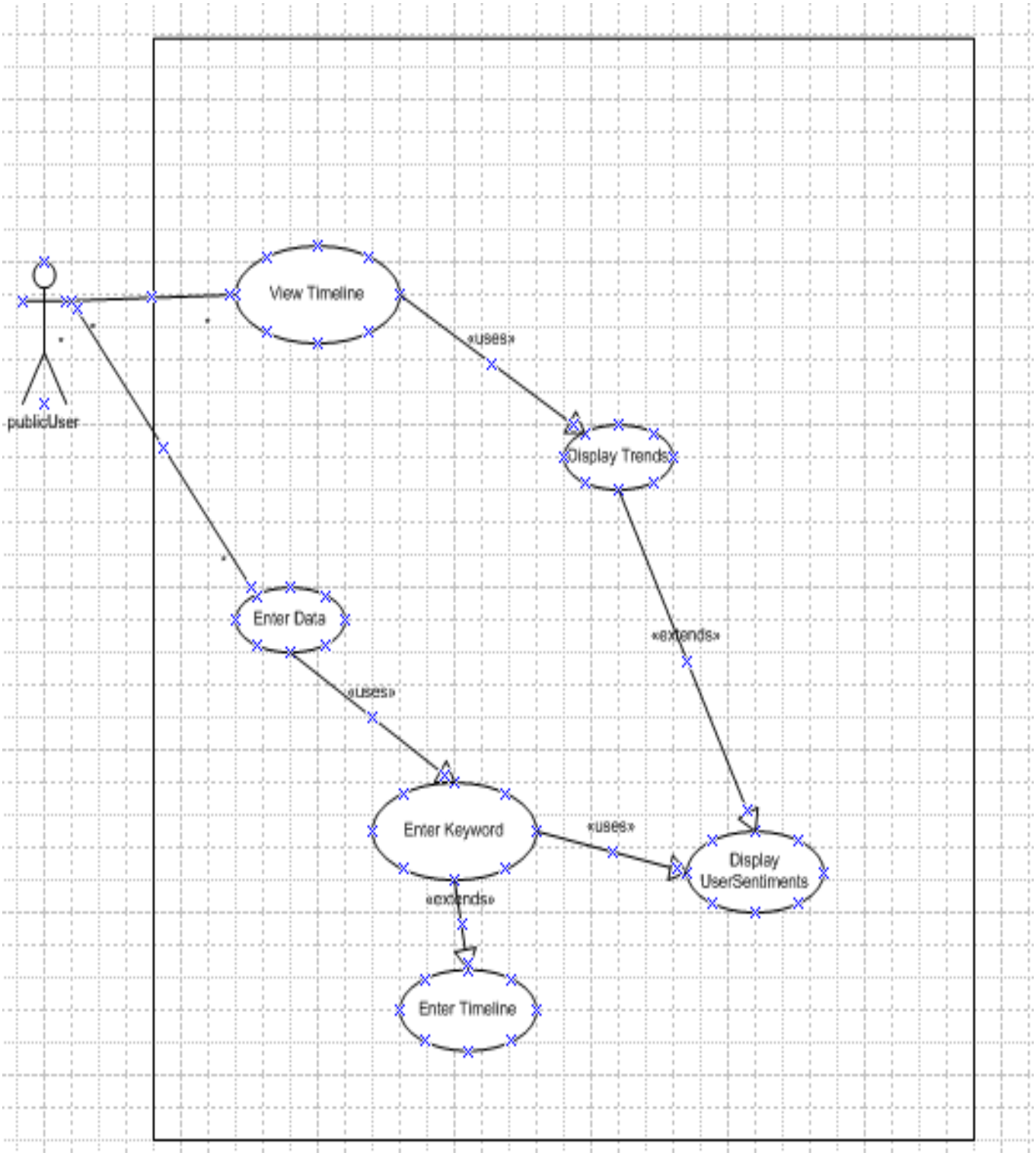

Figure 2-Use Case Public User

3.4.3 Sentiment Analysis

Sentiment analysis will be performed on the user-specified keywords within the text from social media, blogs and news articles to determine the overall mood of the users relative to the topic. The sentiment analysis will provide a negative, neutral, or positive numeric sentiment value.

3.4.4 Performance Requirements

Performance requirements are of vital importance in any project. This section is dedicated to performance requirements related to project.

3.4.5 Real Time Processing

The software must output real time data graphically. In addition, the software may output a graph of mood trends over time, as well as additional statistics pertaining to a topic (average sentiment over all analysis sessions and total number of text from social media, blogs and news articles processed). This output should be clear and easy to understand.

3.5 Non Functional Requirements

Non-Functional Requirements are very critical in success of a project. This section enlists the non-functional requirements for our project.

3.5.1 System Resource Consumption

Most of the functionality is performed at the server end as the main application is based on the backend database. However, some functionality is processed at the client end as well such as responsive and interactive front end using AJAX.

3.5.2Reliability

The software will meet all of the functional requirements without any unexpected behavior. At no time should the graphical output display incorrect or outdated information without alerting the user to potential errors.

3.5.3 Availability

The software will be available at all times on a personal computer as long as the device is in proper working condition. The functionality of the software will depend on any external services such as internet access that are required. If those services are unavailable, the user should be alerted.

3.5.4 Security

User Identification is unknown to the public. Only user conversations will be used to perform the analysis.

3.5.5 Maintainability

The software should be written clearly and concisely. The code will be well documented. Particular care will be taken to design the software modularly to ensure that maintenance is easy.

3.5.6 Portability

The application is a stand-alone web application.

4. DESIGN AND DEVELOPMENT:-

4.1 Introduction

Opinions are one of the basic entities in human conversations. It is in human nature to have feelings and sentiments about the people and the happenings around them. Opinions and beliefs of people around us affect us in determining what choices we make and what do we believe in. Humans generally ask for opinion of other people when they need to make an important decision. Feelings, whether our own or of anyone else, help us in evaluating the world around us. Sometimes, these feelings are positive and refreshing while at other times, they can be negative and hostile. For example, a Pakistani cricket fan will have refreshing feelings when Pakistan wins a cricket match against India but he will be saddened if Pakistan loses the next match.

Human behavior has always been an interesting subject. We, the humans have always wanted to be able to perceive how human nature works. It is important to learn about the existing human behaviors by analyzing how a person feels about a certain event or an individual. The opinions and sentiments of humans are of great value because these emotions actually lay the basis for human behavior.

The following subsections introduce the software design for a Text Analysis Tool, known as UPCYCLER, to its readers. This design document contains architectural level diagrams accompanied by other low level design diagrams. The aim of this document is to enumerate the main components, and describe the relationships and dependencies between them to carry out the various functionalities of the system.

4.2 Purpose of this document

UPCYCLER is a Text Analysis tool which takes text from social media, blogs and news articles in raw form and develop a model to process the natural language. The main idea behind the development of the UPCYCLER is to automate the understanding of apparently useless user conversations. There are multiple real life applications of information extracted from the desired system. Examples include opinion of general public for products and people and threat detection. Theuser conversations are in different forms of languages like English, Roman Urdu and Urdu (Unicode). The system transforms the text into one uniform representation, English and uses the sentiment analysis model to extract the sentiments of users. The results obtained are stored in the database and consequently displayed through a web interface. This Software Design Document enumerates all the components of UPCYCLER. The functionality of each and every module is explained. The dependencies and relationships between different modules or components are also stated in the document. Conventional visual demonstrations from software industry are used wherever necessary. The document plays the role of a reference guide to development of the entire project to its audience. The intended audience includes the following stakeholders:

- Development Team
- Supervisor
- Co-supervisor
- Evaluating Team
- General Public
- Security Agencies

4.3 Scope of the project

UPCYCLER is a text processing tool based on user conversations. The user conversations are composed in various languages like English. There can be multiple representations for a single language. For example, in case of Urdu, Unicode Urdu and Roman Urdu are two representations, that are commonly used in the user conversations. The system at hand uses text from social media, blogs and news articles in English and Roman Urdu in order to develop a natural language processing model.

Depending upon the output from the system, UPCYCLER will be used by two type of users:

- Normal user
- Authenticated user

The system will be used by normal users to review the popular trends and overall sentiments of users about them. Normal users do not need any credentials to use the system by accessing the URL of website. The normal users cannot possibly see the actual user conversations and the associated users. Only the overall sentiments of users are demonstrated in mathematical and visual demonstrations.

The system administrator will maintain a database for authenticated users which will have login credentials. Authenticated users will login on the public website using these credentials. Once, the system authenticates the credentials provided, the users will be redirected to a page with potential threats identified by the system. The authenticated user can get user details for each threat and take further actions as per standard procedures.

Text processing tools can have many potential applications. Due to reasons like precision in the requirements and computational constraints, the scope for UPCYCLER includes the following two applications only:

1. Sentiment Analysis
2. Threat Detection

Sentiment Analysis accounts for the emotions involved in the user conversations. This is possible by assigning scores to keywords which represent some sort of feelings. First of all, raw user conversations are transformed into one uniform language, English. Sentence structure is then extracted to identify the subjects and the associated adjectives. The adjectives for each subject are classified for sentiment analysis. The broader classification for emotions is positive or negative. Further classification for representing the level of positivity or negativity is done using scores in numeric values. This application targets the general public by displaying the popular topics and the sentiments of people about these topics.

Threat Detection is another useful application of language processing. It is impossible to go through every message looking for threats. Therefore, it is reasonable to automate the threat detection procedures. Even if, the automated system cannot detect potential threats accurately, it will definitely reduce the number of user conversations significantly, so that it is possible for humans to confirm if a message is threatening or not. Once, a model has been developed for understanding user conversations , it will be used for identifying the potential threats. Threat Detection focuses on the negative sentiments from the models and a threshold value will be used to classify a message as a possible threat. All possible threats are notified to authenticated users upon login to the system.

4.4 Definitions, Acronyms, and Abbreviations

Here are a list defining the difficult terms and abbreviations used in the document:

- **UPCYCLER:** The name of the project comes from the term Up cycling. Up cycling is the process of converting useless things or data into useful information.
- **NLP:** Natural Language Processing is the process of performing computations on data in form of natural language in order to make it understandable by machines.
- **Sentiment Analysis:** Sentiment Analysis is the process of analyzing the opinions of users, grading them as positive or negative, and assigning scores based on intensity of emotions.
- **SQL:** Structured Query Language
- **NLTK:** Natural Language Toolkit
- **GUI:** Graphical User Interface

4.5 Overview

This section gives an overview of the remaining sections in this document. Section 2 comprises of system architecture description. It contains the overview of working modules. Complete overview of the project is demonstrated using a complete diagrammatic view of the system. A high level design diagram is also used to visualize the structure within the system as well as individual modules. State machine diagrams are also used to show the basic workings in the project. Low level states using state machine diagrams are given with backend processing perspective as well as users' perspectives.

Section 3 has detailed description of each module in the system. Each sub module is described as well. Diagrams are used wherever necessary for better understanding. Sketches for output screens are included in the subsection of web interface. Section 4 covers low level design diagrams like use cases, ER diagrams, sequence diagrams, and activity diagrams. Section 5 contains information about reusable components in the system. There are two subsections in this section. One section has information about components that have been reused in the system while the other focuses on the reusability of components developed in the project. Section 5 has information that explains the design decisions taken in development of project. It also explains the tradeoff between two approaches to web interface development. Section 6 has pseudo code for modules and their interfaces.

4.6 System Architecture Description

This section covers the brief discussion of each module and its relationships with the other modules in the system. Graphical representations are used for better understanding as well.

4.6.1 Overview of Modules

Based on functionality, there are four main modules in the UPCYCLER project. Section 3 has detailed description of each module including definition of subcomponents and their functionality. This section briefly states the overall functionality of each module in the system. The four modules are as follows:

- **Text Processing Module:** This component of the system makes use of NLP techniques and extracts the structure of sentences. Discussion topics and the adjectives associated with them are extracted and passed to sentiment analysis module for further computations.
- **Sentiment Analysis Module:** Sentiment Analysis Module uses the output from the text processing module. Sentiment Analysis algorithms are then used to classify each sentiment as positive or negative. Scores are also assigned to represent the intensity of feelings. The results are stored in the databases.
- **Database Module:** The database acts as a bridge between the backend processing modules and the frontend web interface. The topics of discussions, associated sentiments, users and threats are stored in the database.
- **Web Interface Module:** The web interface retrieves data from the database and updates the output screens. There will be at least one common welcome screen and different output screens depending upon the type of users. Here is a diagram showing the complete diagrammatic view of the project using the modules, their sub modules and the relationships to show the flow of the system.

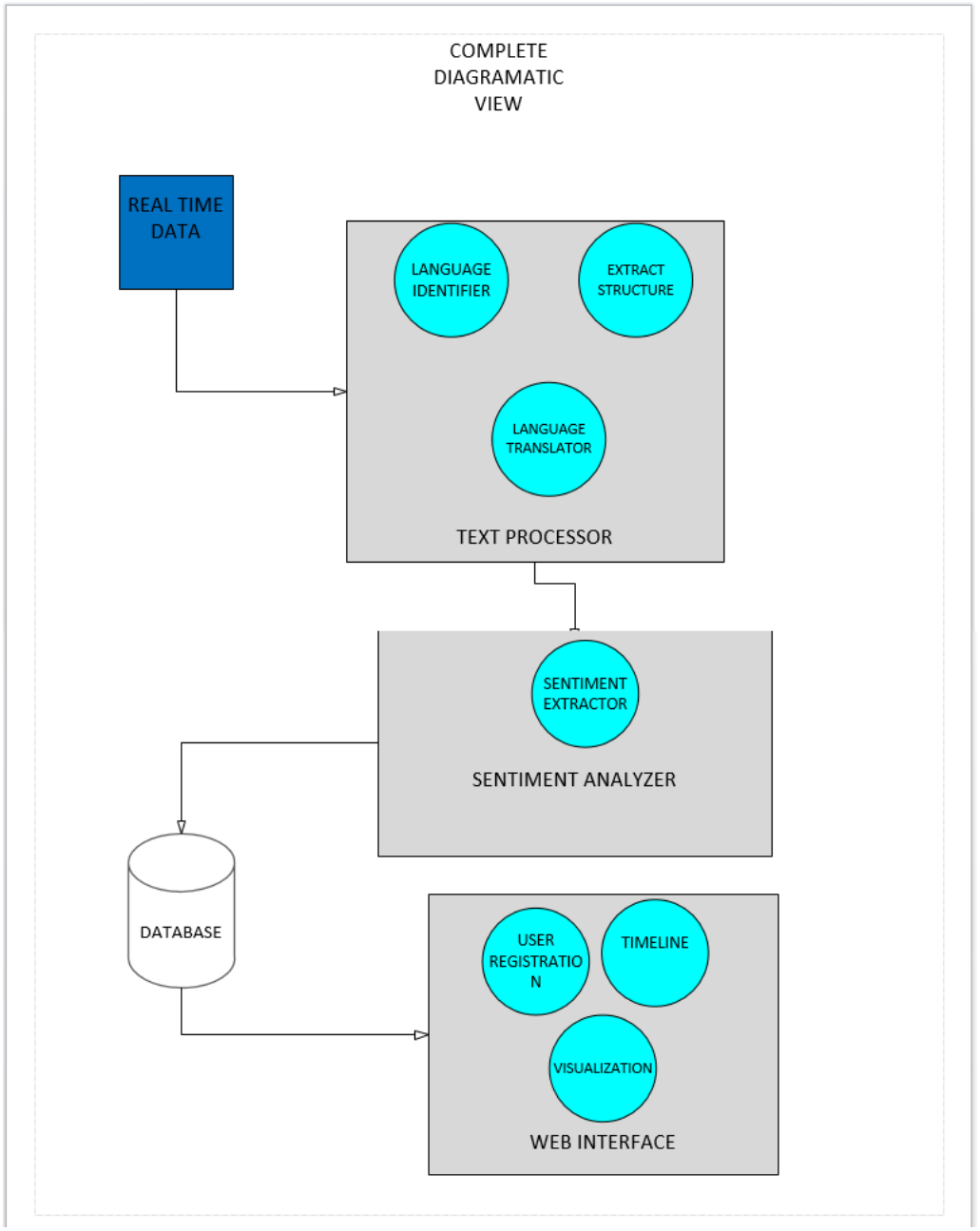

Figure 3- Complete Diagrammatic view

4.6.2 Structure and relationships

Each module in the system is properly interfaced for communication with other modules in the system. The modules are interconnected and depend upon each other for complete functionality. For example, Sentiment Analysis module cannot start processing unless Text Processing module has passed some structured results to it. If Sentiment Analysis module has not processed any user conversations, database will not have any entries and consequently, the web interface would have nothing to display. Hence, the

functionality from each module is partially dependent on other modules. The various modules and the sub modules are displayed in a structural format using the following high level design diagram.

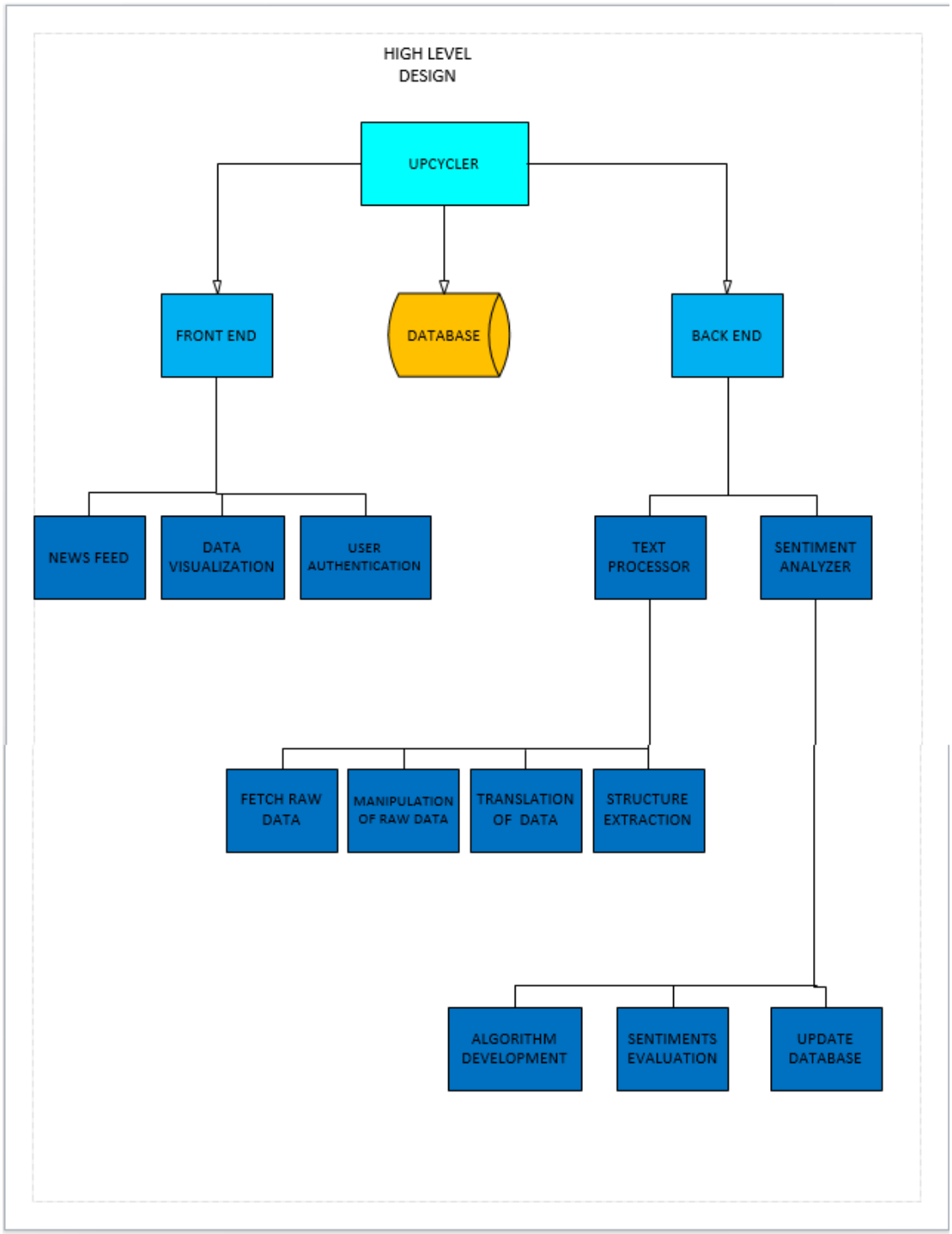

Figure 4-High Level Design

Following two subsections contain three state machine diagrams which shed some more light on the structure and relationships between components. For the sake of simplicity, backend processing and frontend interface have been demonstrated in separate diagrams.

4.6.3 State Machine Diagram (backend)

The main model of computations is constructed at the backend and involves the two modules, Text processing module and sentiment analysis module. Database module is also used for storing the results from sentiment analysis module.

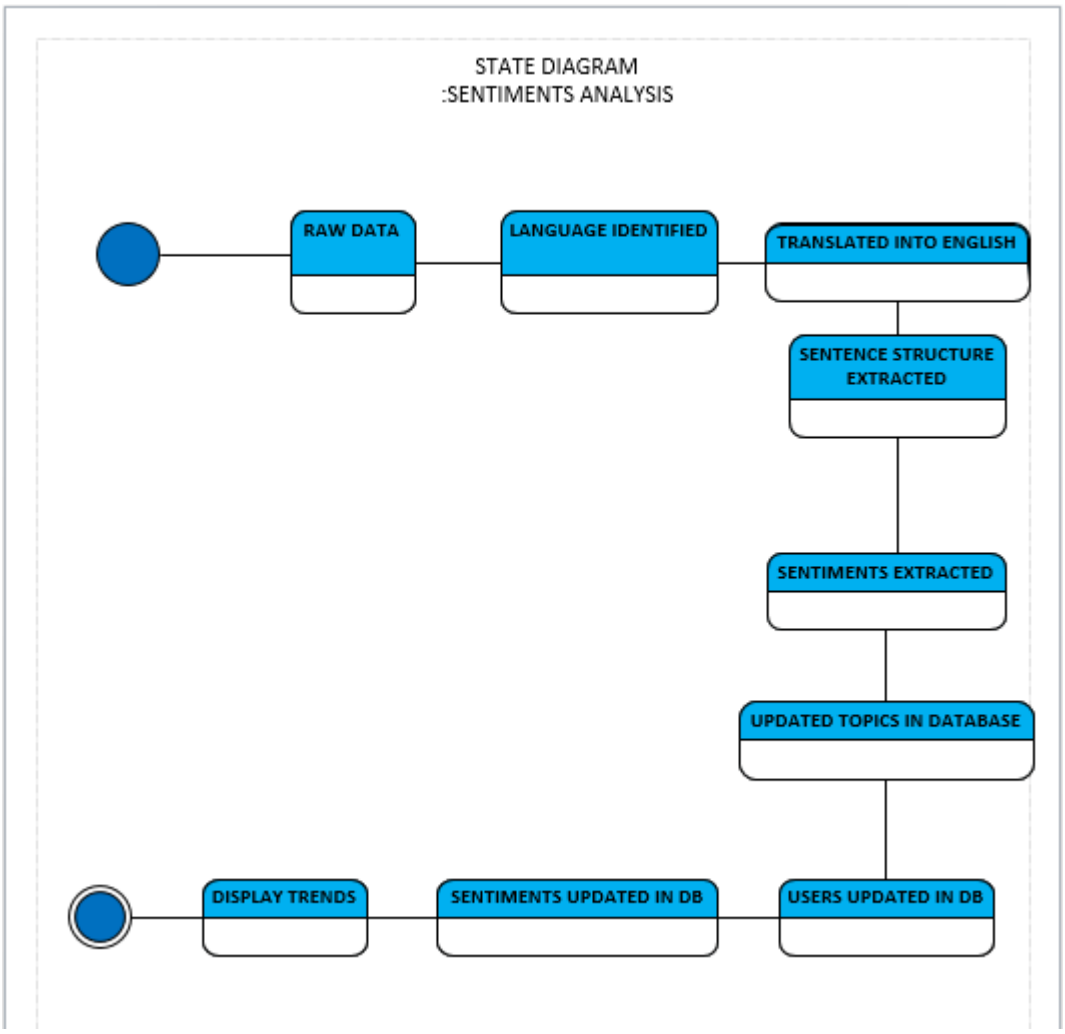

Figure 5-State Machine Diagram

The state machine diagram in above figure demonstrates the transition of states to carry out the backend processing. Backend processing goes through each state to establish

the model which will be used by web interface. Let us describe functionality with respect to each state:

1. **Raw Data:** This state represents that raw textual data has been retrieved. The user conversations can be in any representation of any language.
2. **Language Identified:** This state is achieved by identifying the language used in each text message. Each message is categorized as text in English, Roman Urdu or other languages. English and Roman Urdu user conversations are used for further computations. User conversations from other languages are simply ignored.
3. **Translated into English:** This state comes after user conversations in Roman Urdu are translated into English. This is accomplished using a Roman Urdu to English dictionary.
4. **Sentence Structure Extracted:** The next state transition comes in when sentence structure is extracted for all the text in English whether translated or not. The sentence components like nouns, adjectives and verbs are identified.
5. **Sentiments Extracted:** This state is achieved after the core algorithms for sentiment extraction have been applied. Each message with expressions involved are scaled for the measure of negativity in it. Numbers are used to represent the intensity of the opinions. A threshold value is used with the intensity values for identification of threats.
6. Once the sentiments have been assigned scores, topics, users and scores are updated in the database in the next three states. The final state in the backend processing is when the database has been populated.

4.6.4 State Machine Diagrams (frontend)

Two state machine diagrams are used to represent the states normal users and authenticated users go through when using UPCYCLER.

The following diagram shows the state transitions for normal users.

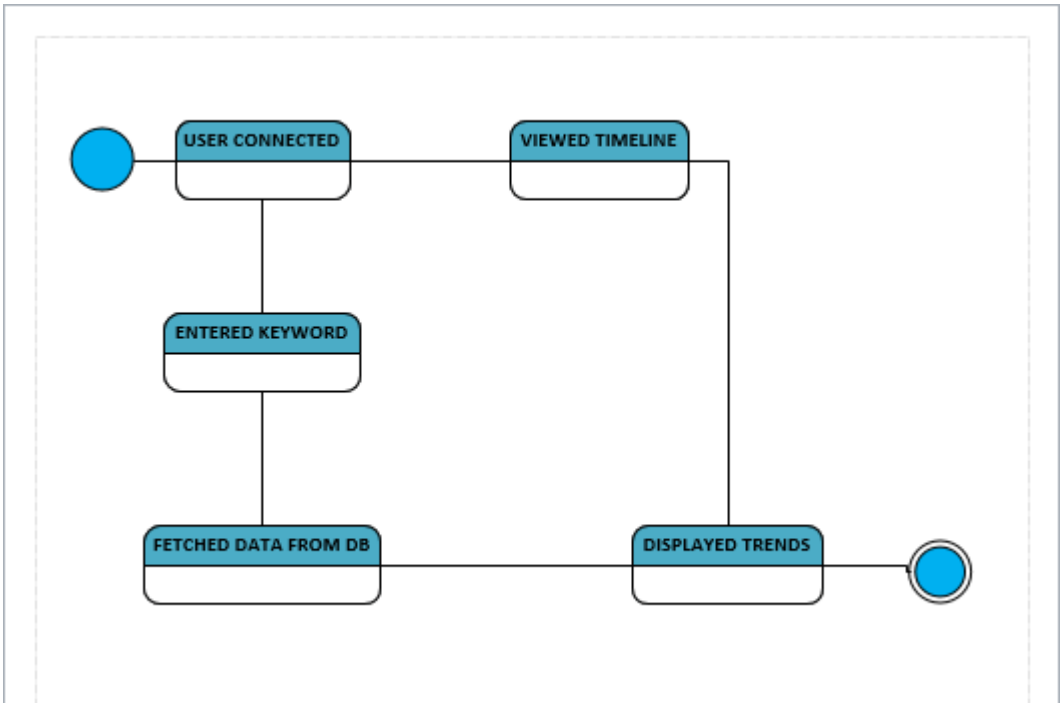

Figure 6-State Machine Diagram (Frontend)

1. **User Connected:** User is in this state after he has selected the public user option from welcome screen.
2. **Viewed Timeline:** This is the default state user goes to after selecting the user type.
3. **Entered Keyword:** This state is based on a search query. User is in this state when he wants to search sentiments about a particular topic.
4. **Fetched Data:** This state is achieved after querying the database. This state basically represents the time taken for results of a query to be retrieved from the database to web interface.
5. **Display Trends:** This is the state where sentiments of users are visualized to the users. This state is the final state for both paths given in the diagram. Final output is graphical representation of trends and the associated sentiments. The following diagram shows the state machine diagram in case of authenticated

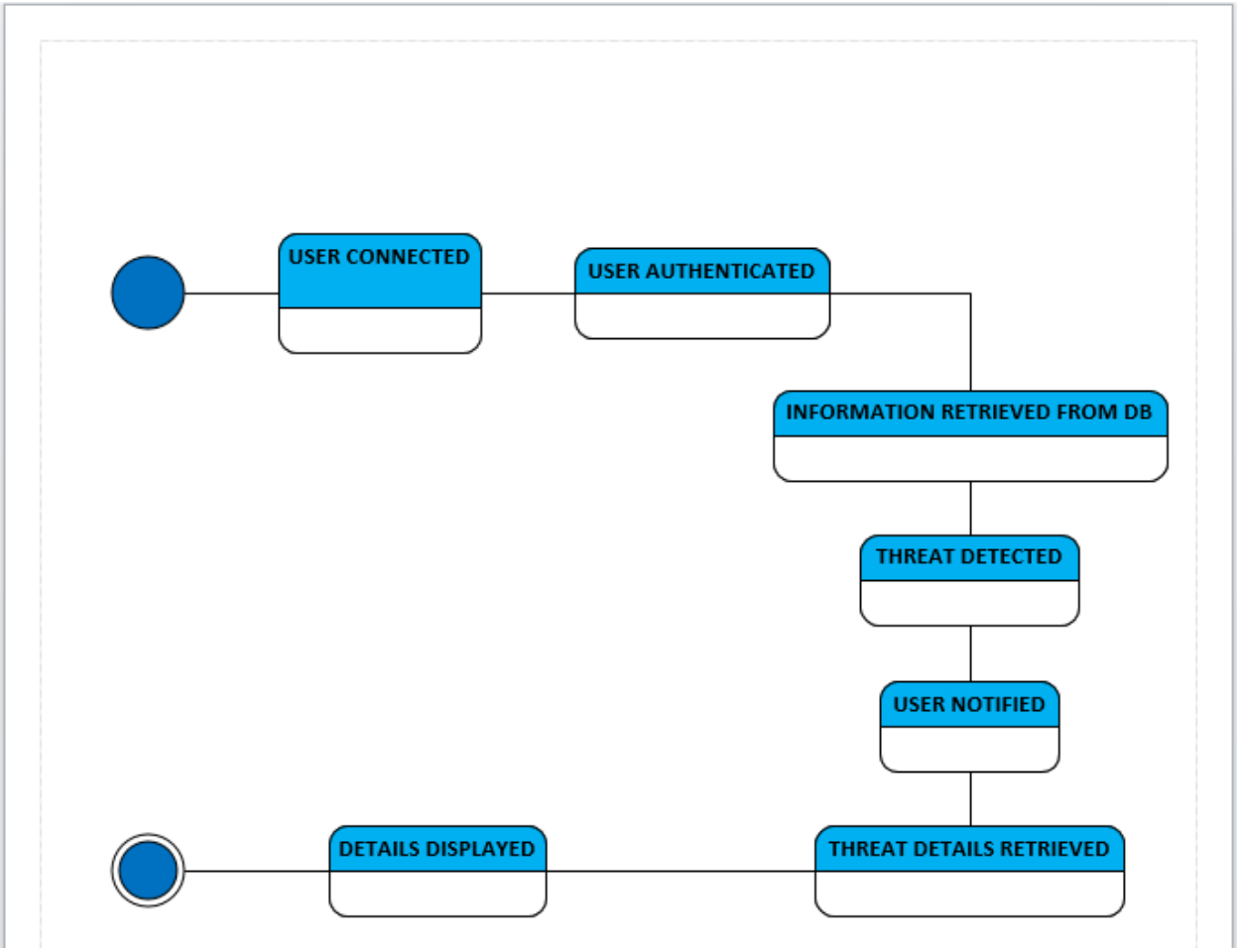

Figure 7-State Machine Diagram (Frontend)

User .Here are the description for each state involved in the above diagram:

1. **User Connected:** User is in this state after he has selected the authenticated user option from welcome screen.
2. **User Authenticated:** This state is achieved after user has login successfully by entering valid credentials.
3. **Information Retrieved:** The state representing the time taken for retrieving information from database.
4. **Threats Displayed:** When threats have been retrieved from the databases, they are simply displayed to user through web interface in this state.
5. **Threat Details Retrieved:** User confirms a message as threatening and directs the system to retrieve further information from database about the message.
6. **Details Displayed:** Details from query in the previous state are displayed in this state.

4.6.5 User Interface Issues

Web GUI is used for interaction of users with the system. There are at least 4 screens involved for both types of users. The home screen is the welcome screen, which has only two buttons. The two buttons are used to select the type of user. Normal users are directed to the public sentiment analysis page where trending topics are displayed along with the public sentiments. Authenticated users are taken to login screen. Once, the user has entered the valid credentials, the authenticated user is directed to threat detection page. Section 3.4 has detailed description for each of these output screens along with visual demonstrations.

The text processing tool passes the results from its computations to sentiment analysis module. The sentiment analysis module uses these results for extracting the sentiments of users. The database module is then populated with results from Sentiment Analysis module. Web GUI extracts information from database and displays it to the users. Web Interface and Database modules also interact for authentication of users. Normal users and authenticated users share the same database for retrieving the required information on output screens.

4.7 Detailed Description of Modules

This section comprises of detailed descriptions of each module. The sub modules of each module are described as well. The inputs and outputs associated with each module are discussed as well. Interfaces between the modules, however, have been demonstrated in the section 2.2.

4.7.1 Text Processing Module

Text Processing Module deals with the textual data in raw format. Raw user conversations can be in multiple languages and each language may have multiple representations. The module identifies and uses the text in English language and converts text in Roman Urdu format into English as well which is then used for sentiment extraction by Sentiment Analysis module

This module performs following major tasks:

1. Identification of the language used in user conversations .
2. Translation of Roman Urdu into English.
3. Extraction of sentence components like subject, adjective and verbs.
4. Passing the results to Sentiment Analysis module.

To keep the functionality modular, the various functionalities of the module are divided into multiple subcomponents. The following subsections describe the subcomponents which work together for complete functionality of this module.

Identification	Text Processing Module
Type	A module
Purpose	The raw data is in multiple languages and representations. Text processing is necessary to transform text into one uniform representations.
Function	This module transforms the raw data into English and then extract the sentence structure.
Subordinates	Language Identifier, Translator, and Sentence Structure Extractor.
Dependencies	Raw data is taken as an input. The outputs from module are used as inputs to sentiment analysis module.
Interfaces	External Interfaces are required for interaction between the module and sentiment analysis module. An external interface is required for retrieving the raw inputs as well. Internal interfaces include the message passing between Language Identifier and Translator components for transforming text into one uniform representation.
Resources	Resources required include the raw user conversations with user identifiers. Text processing demands fast CPU execution times for efficiency.
Processing	Processing takes place in three steps. The steps are language identification, translation of Roman Urdu into English, and sentence structure extraction. Pseudo code for the module is given in section 7.
Data	Raw textual data is in multiple languages. Roman Urdu and English user conversations are used for development of model. Roman Urdu text is also translated into English as well. The output from the module is the data in structured format.

4.7.1.1 Language Identifier

The raw user conversations can be in any language. The scope of the project only includes user conversations in English. The user conversations are categorized as either English or other languages. Conversations from all other language

representations are ignored. Conversation in English language can be identified by simply using an English dictionary. If all the words in a sentence belong to the dictionary, the sentence is categorized as an English sentence. The categorized sentences are then used for further computations by other components of the Text Processing Module.

4.7.1.2 Sentence Structure Extractor

This component makes use of the outputs from other two components in the current module. The English text obtained from language identifier and translator are processed and sentence components like nouns, adjectives, and verbs for each sentence are extracted.

4.7.2 Sentiment Analysis Module

This module carries out the core functionality of the project, sentiment analysis. The input into the system is the structured textual data in English language and extracts the sentiments from conversations. The results are stored in the database.

Identification	Sentiment Analysis Module
Type	A Module
Purpose	To carry out the main functionality of the system by extracting sentiments from user conversations.
Function	Extract the sentiments from user conversations and detect threats among them.
Subordinates	Algorithm Development, Sentiments Evaluation, Threat Detection, and Database updating.
Dependencies	The module depends directly on the output from text processing module as the algorithms will work with structured data received from the text processing module.
Interfaces	External interfaces are required to receive input from text processing module and save the outputs in the database. Internal interfaces are developed for communications between the subordinates of the module.

Resources	Resources required for this module to work are only the outputs from text processing module.
Processing	There are different components for Sentiment Analysis and Threat Detection. Pseudo code for the whole module is included in section 7.
Data	The input data is in structured format and the output data comprises of topics of discussion, sentiments and threats. The outputs are saved in the database using proper interfaces.

Following subsections cover the subcomponents which work together to complete the functionalities of this module.

4.7.2.1 Algorithm Development

This component is dedicated to the development of algorithm for evaluating the opinions contained in the user conversations. This algorithm is used for sentiments evaluation. The inputs to the algorithm are the user conversations in structured format and the outputs are scores indicating the level of sentiments involved in them.

4.7.2.2 Sentiments Evaluation

This component assigns scores the intensity of emotions encoded in the user conversations. The scores are assigned in numeric form. The scale can be labeled as the negativity measure as the lowest number is ranked as most positive while the highest number is rank as the most negative sentiment.

4.7.2.3 Threat Detection

This component uses the sentiments extracted and uses a threshold value to detect the conversation that could be possible threats. The higher intensity of negativity is considered as threatening, and a separate dictionary can also be developed for detection of threats.

4.7.2.4 Updating the database

This component plays the role of an interface from Sentiment Analysis module to the database module. The results from computations are stored in the database. Entities like users, topics of discussions, threats, sentiments, and others are updated.

4.7.3 Database Module

Database module is designed to act as a bridge model between the backend processing modules and front end graphical web interface. The information to be displayed is stored in the databases after backend processing. The database is accessible by the web interface for tasks like user authentication, data visualizations, and threat detection.

Identification	Database Module
Type	A module
Purpose	The output from sentiment analysis module has to be saved in a database, from where it can be uploaded onto the web interface.
Function	The core functionality of this module is to design a database solution for the application. Proper ER diagrams are used for designing the solution and interfaces are provided for interaction with the modules used for inputs and outputs.
Subordinates	Entities and their attributes along with the relationships.
Dependencies	The database is populated with results from sentiments and threats identified from sentiment analysis module. Hence, there is a direct between these two modules. The web interface is dependent upon this module as the database acts as an information model.
Interfaces	External interfaces are required for interactions with web interface and sentiment analyzer. Internal interfaces are simpler as database solution has already been devised as represented through the ER diagrams in section 4.2.
Resources	Resources required for this module are an SQL database server.
Processing	Pseudo code for the module is given in section 7.
Data	Data is stored in tables and are accessible through web interface.

The database will comprise of following main entities:

- Authenticated Users

- Users
- Raw user conversations
- Structured data with sentiments
- Topics
- Threats

Authenticated users have login details which have to be stored in the database. The web interface queries this entity for validation of users. New entries can only be entered manually by the system administrators.

Users are the users whose user conversations are being used in the development of working model. It will have attributes like user ID, username, and location. Each message will be associated with two users, one as sender, and the other one as receiver.

Raw user conversations are the unstructured sentenceswhich were in English or had been translated into English by text processing module. It is important to store them so that authenticated users can read the actual user conversations instead of structured information as the system cannot possibly guarantee 100 percent accuracy.

Structured data with sentiments, topics of discussion, and threats are the output from Sentiment Analysis module. Each of these entities is stored in the database to be accessible by the web interface. The web interface does not retrieve all the information from the database. Only desired information is retrieved from the database to minimize the bandwidth requirements. In case of public user, desired information includes the most trending topics and the sentiments of people about them. If public user enters a keyword in the search column, the relevant information only retrieves the sentiments about topics that are matched with the keyword. In case of authenticated user, the desired information is retrieved only from threats section and users table.

4.7.4 Web Interface Module

This module is the only module that interacts with the users directly. The web interface comprises of a website that can be accessed by public users and authenticated users. Website is the frontend module in the system, which is used to visualize the output from the two backend modules, Text Processor and Sentiment Analyzer via the Database module. This is demonstrated in the overall architecture of the system diagram as well.

Identification	Web Interface Module
Type	A module

Purpose	This module is for the users to interact with the application.
Function	Develops a web interface which displays the output from backend processing to the users via a central database.
Subordinates	Multiple pages or screens.
Dependencies	This module directly uses the database to access the computational results. Database is updated by the two backend processing modules.
Interfaces	External Interfaces include the procedures for extracting information from databases and the webpages for interacting with the users for inputs and outputs. Internal Interfaces include how different webpages are interconnected.
Resources	Resources required for this module are a web server that can access our database module.
Processing	Pseudo code for this module is given in section 7.4.
Data	The data required by the module is contained in the database. The database has to be populated before the application is uploaded. User inputs taken at runtime such as search queries and getting threat details also interact with the database.

There are at least four pages involved for both types of users to complete the functionality of desired system. However, more pages may be added in the future. Following figures represent the sketch for various screens involved in the system.First of all, all users are taken to the welcome page. The user type is selected by user using the available choices on the page and are redirected to different pages depending upon their choice.

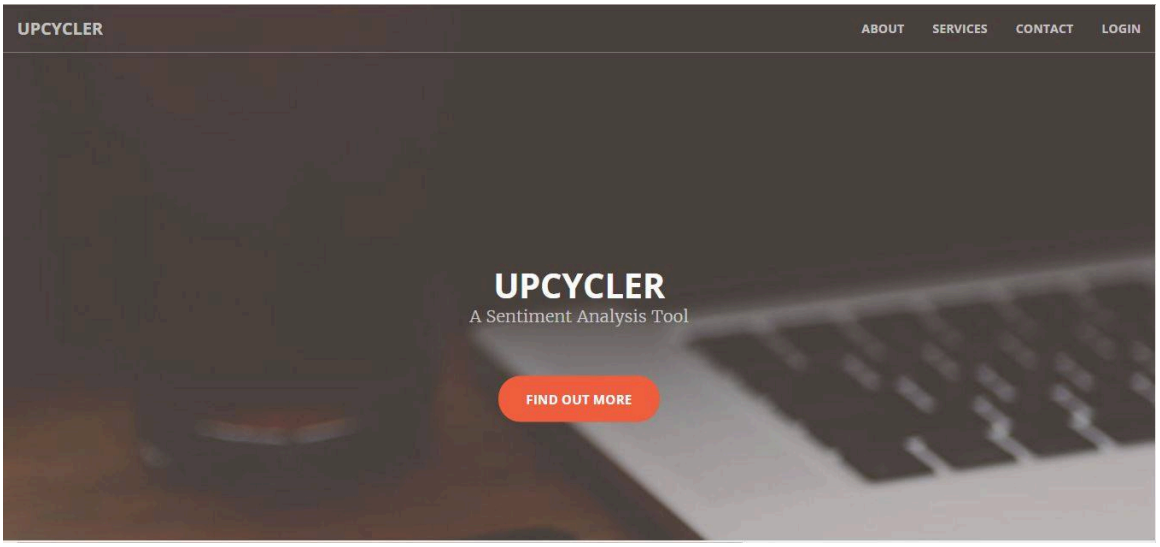

Figure 8-Screen 1 : Welcome Page

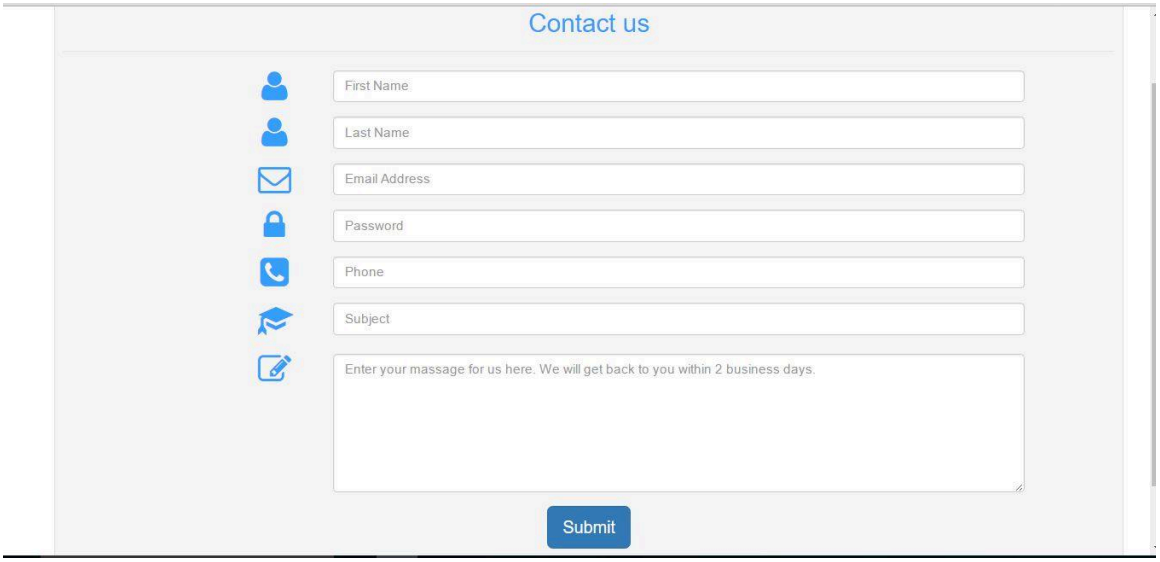

Figure 9-Registration Page

Figure 10-Sentiment Analysis

The screen above represents a sketch for how the output screen for normal users look like. The public users from welcome screen are directed to the page dedicated for sentiment analysis. This page is used for visualization of popular topics of discussions and sentiments of people about them. If we take the screen shown above as an example, there is a trending topic, Metro bus, and sentiments of people are visualized. The x-axis displays the various scales of negativity while y-axis is used for representation of percentage of users who gave an opinion about the trending topic.

Please note that this is just a sketch of actual output screen. Unlike the above demonstration, which has only one topic of discussion, actual output screen will have multiple discussion topics. The main principle for representation of sentiments will remain the same in the actual output screens with possible slight variations.

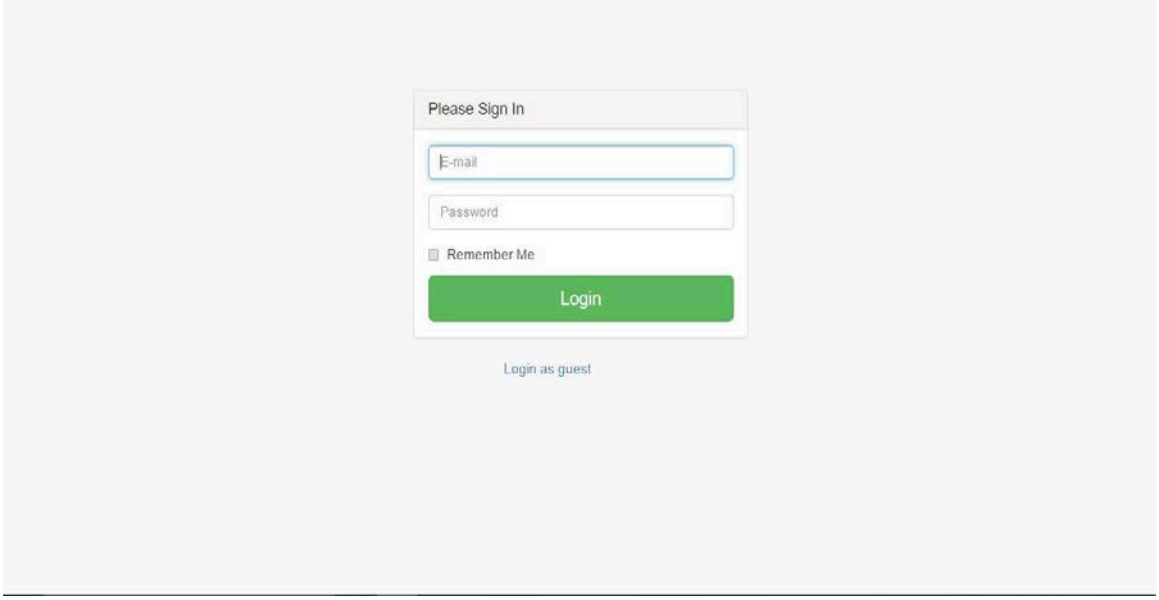

Figure 11-Screen 2b: Login

Authenticated users from welcome screen are directed to screen 2b, the Login page. The users have to enter a username and password and click the login button to request validation of credentials. Once the credentials have been validated, the authenticated users are taken to Screen 3.

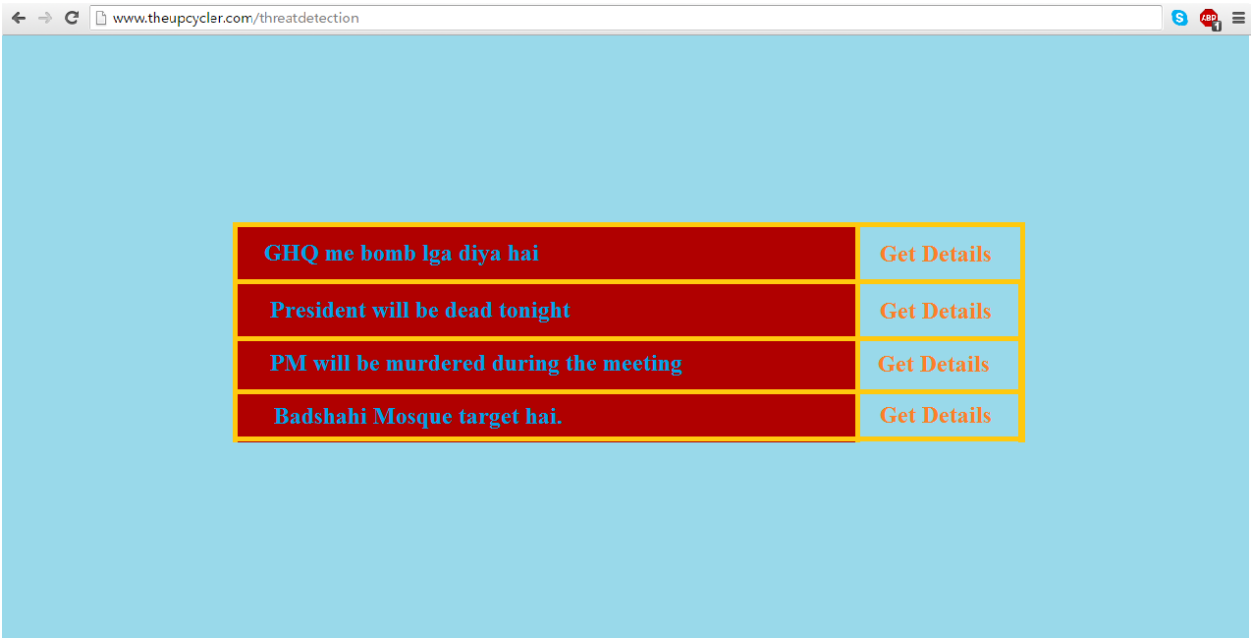

Figure 12-Screen 3: Threat Detection

Screen 3 demonstrates a sketch for the output screen for authenticated users. The page is dedicated to threat detection. The user conversations which could be possible threats are listed. Authenticated users can skim through each message and if he sees any particular message as threatening, and can get details for it like Sender ID, Receiver ID and time. Future actions of authenticated users are determined by standard procedures of agencies or personal discretion. Please note that these screens only represent a sketch of actual output screens to give a basic idea of the web based user interface. More details will be added in the future.

4.8 Detailed Design

4.8.1 Use Cases

This section encompasses the use cases of the system. Depending upon the users of the system, there can be two main use cases. These are described in the following subsections.

4.8.1.1 Analyze Sentiments

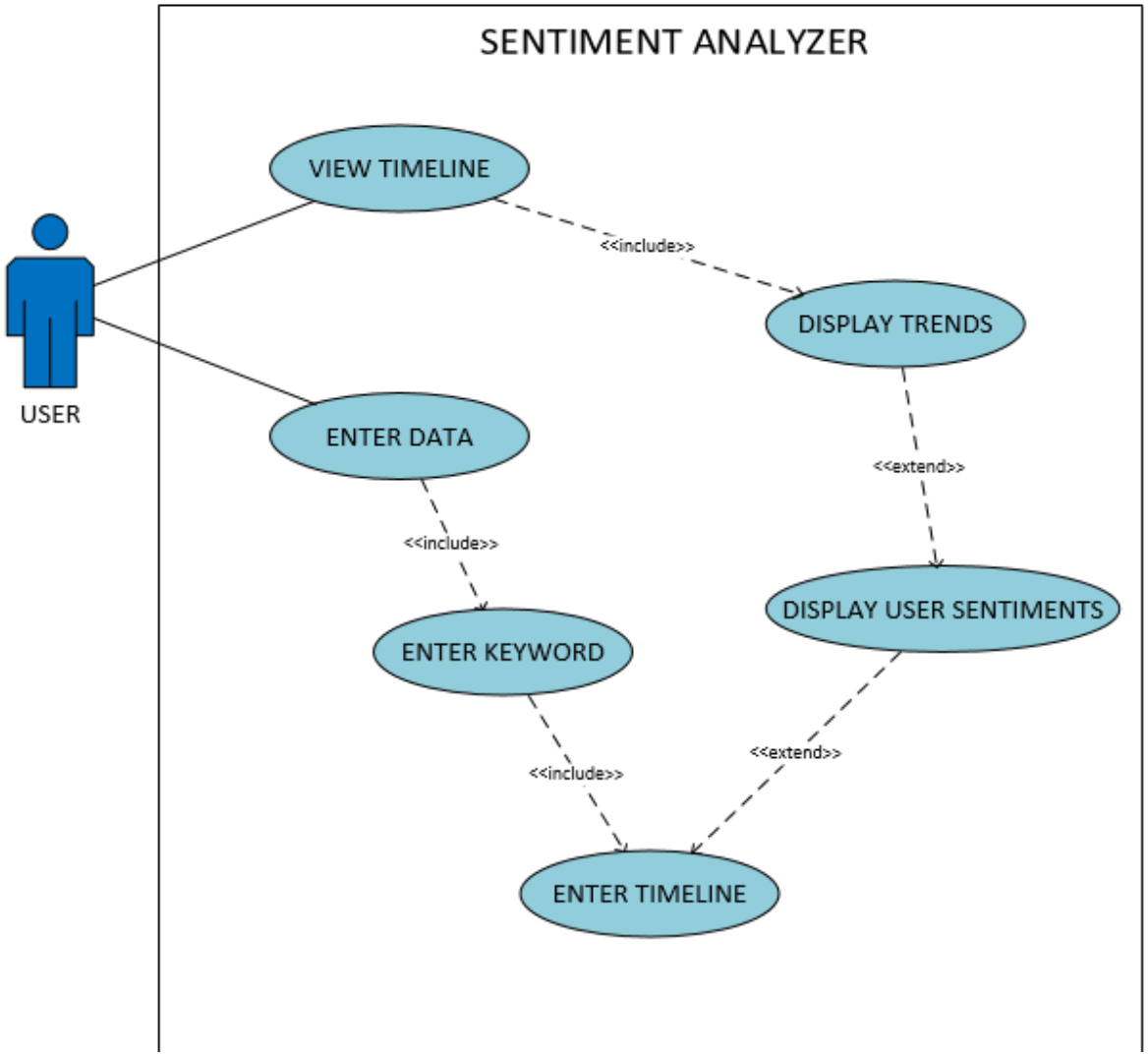

Figure 13-Usecase 1 -Analyze Sentiments

This use case is aimed for the public users which only analyze the sentiments of users about certain topics. The following table has further description about the use case.

Use Case Name	Analyze Sentiments
Actor(s)	Public Users

Pre-Condition	Public user has stable internet connection and has access to the website.
Normal Course	User views the timeline showing the popular trends, and analyzes the sentiments of users.
Post-Condition	User is aware of public sentiments about trending topics.
Alternative course	User enters a keyword in the search box. Topics and the associated sentiments, relevant to the keyword are retrieved.
Post-Condition	User becomes aware of public sentiments about the searched topic(s).
Priority	High
Frequency	High

4.8.1.2 Detect Threats

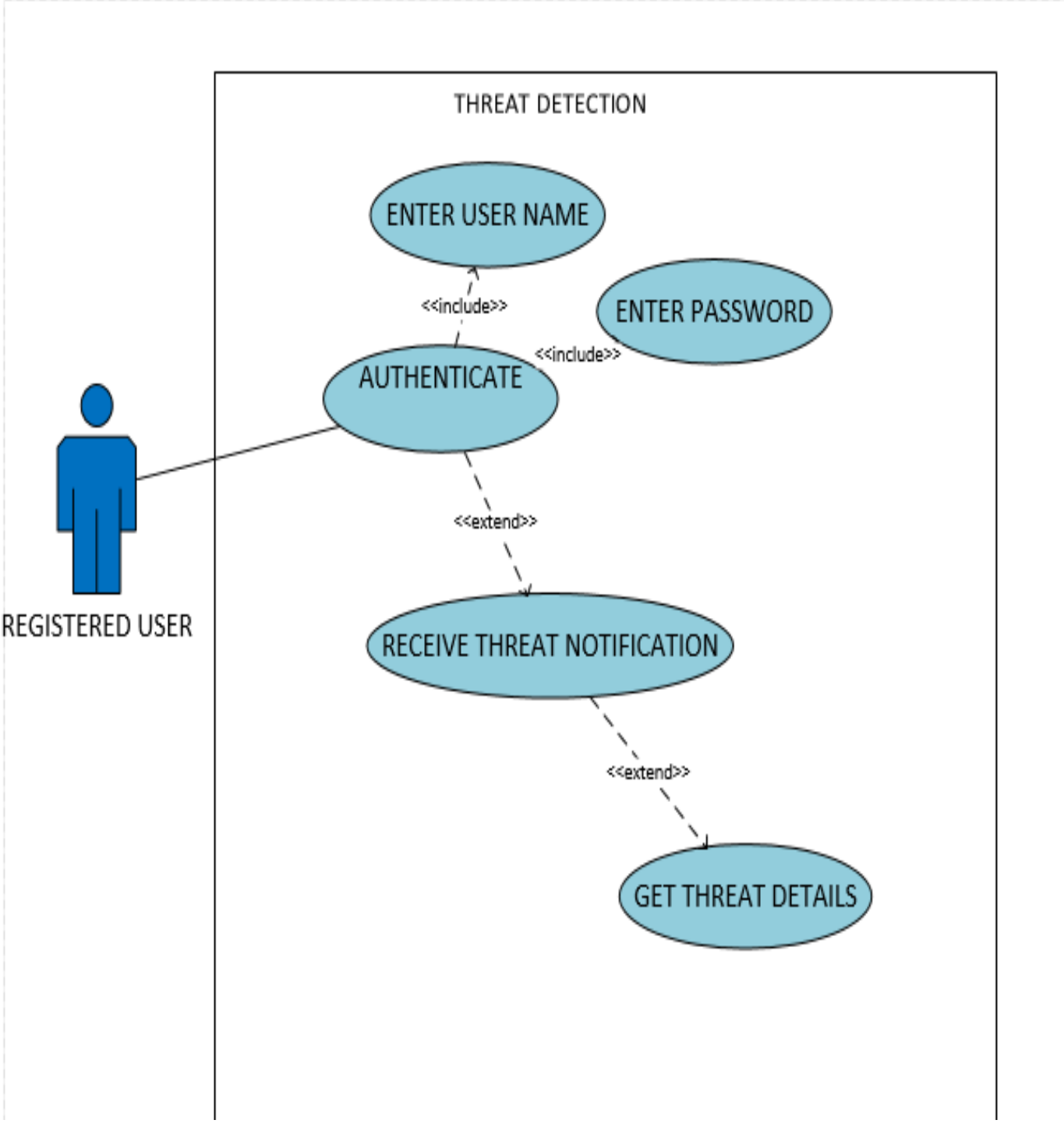

Figure 14-Use Case2: Detect Threats

This use case covers the second application of the project, UPCYCLER. It demonstrates the steps taken for detection and confirmation of threats using the system. The following table has further details about the use case.

Use Case Name	Detect Threats
Actor(s)	Authenticated Users
Pre-Condition	User has a stable internet connection and selects the authenticated user option from welcome screen.
Normal Course	User enters his credentials. Authentication is requested over the internet. Once the authentication process is complete, user can review the possible threats. User can get details like sender and receiver ID for potential threats.
Post-Condition	Authenticated user is notified of all the possible threats.
Alternate Course	In case of invalid credentials, the process is restarted at the authorization phase.
Post-Condition	Error message is displayed.
Assumptions	Nil
Priority	High
Frequency	Low

4.8.2 ER Diagram

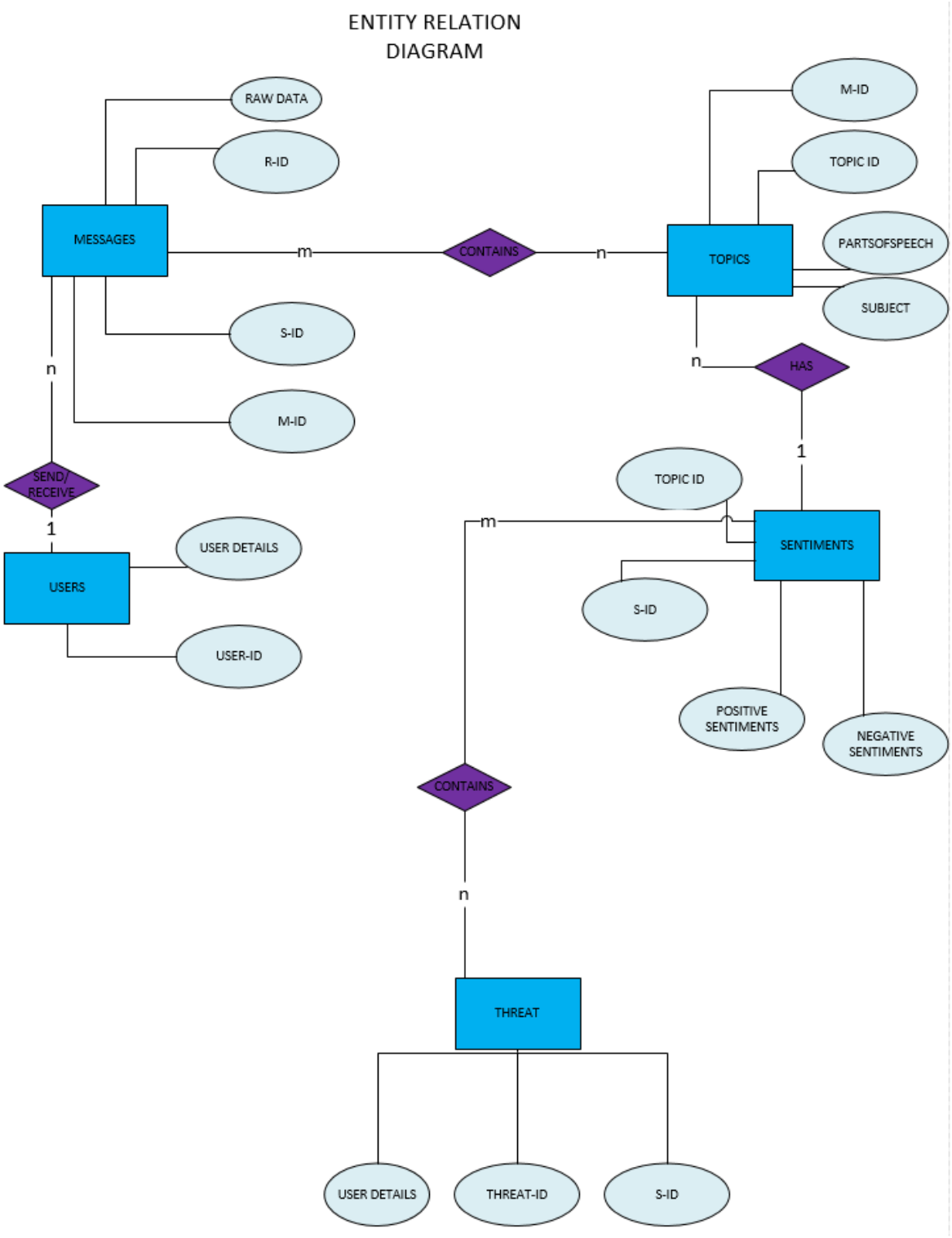

Figure 15-ER Diagram

4.8.3 Activity Diagrams

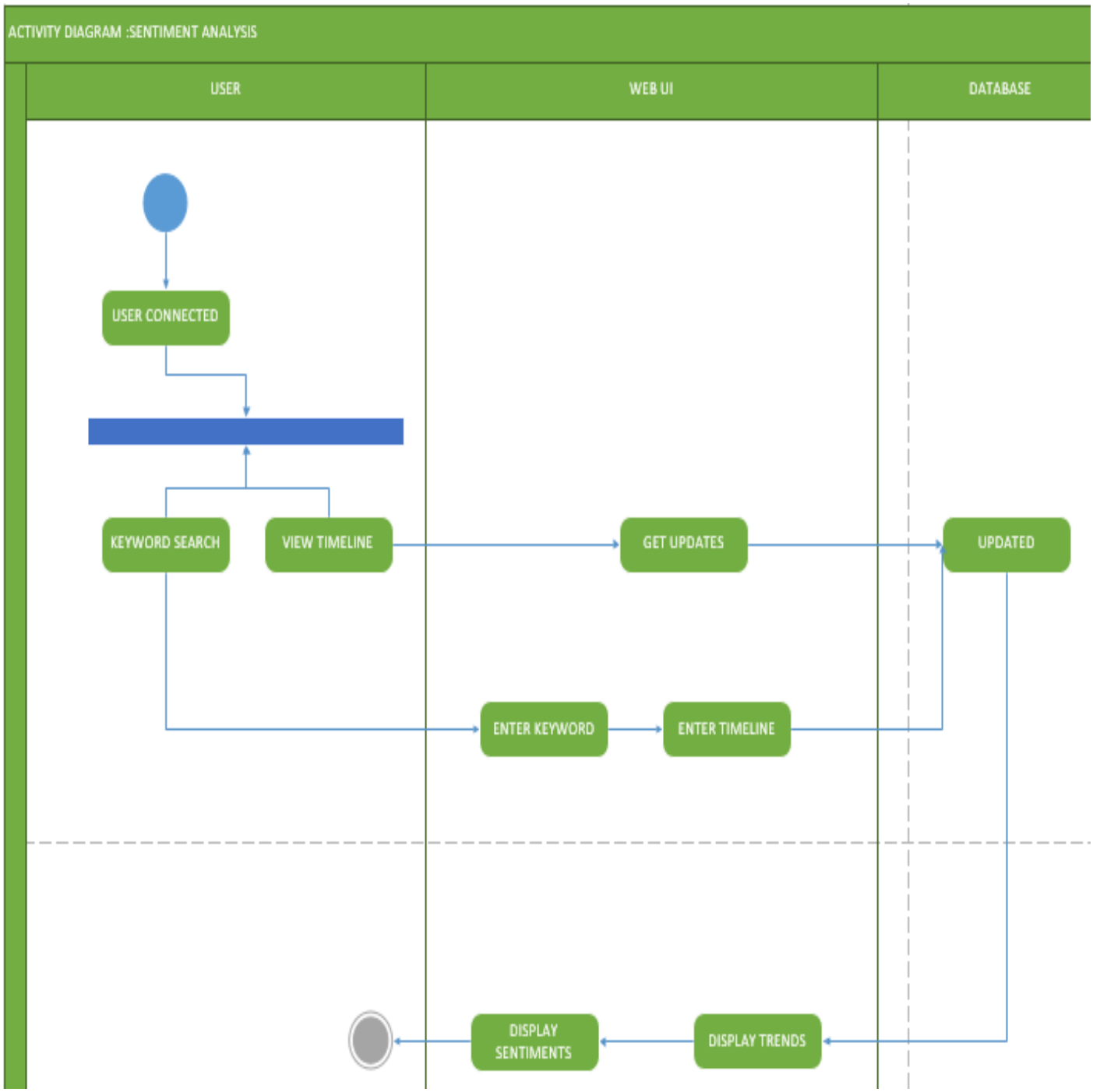

Figure 16-Activity Diagram 1: Sentiment Analysis

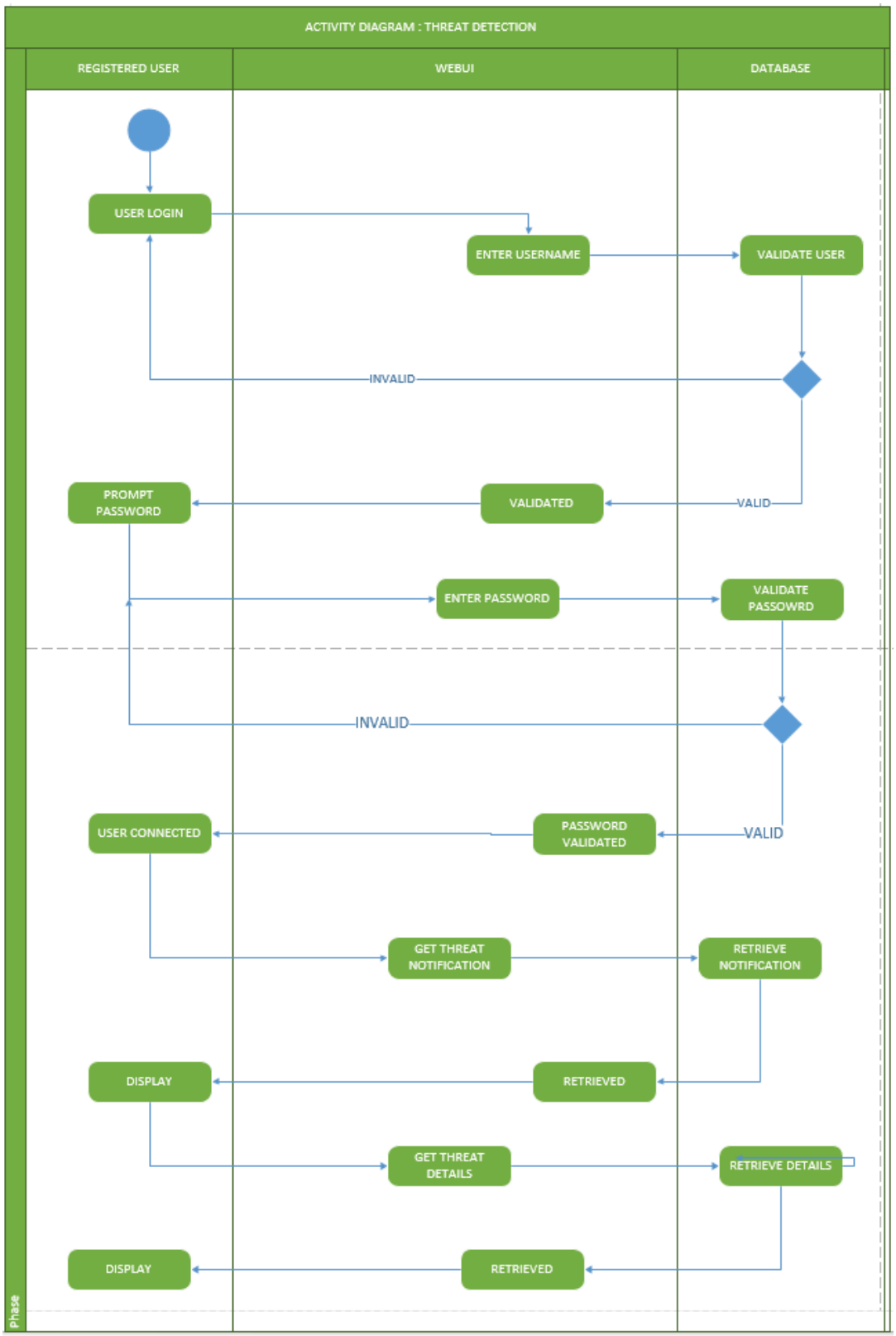

Figure 17-Activity Diagram2: Threat Detection

4.8.4 Sequence Diagrams

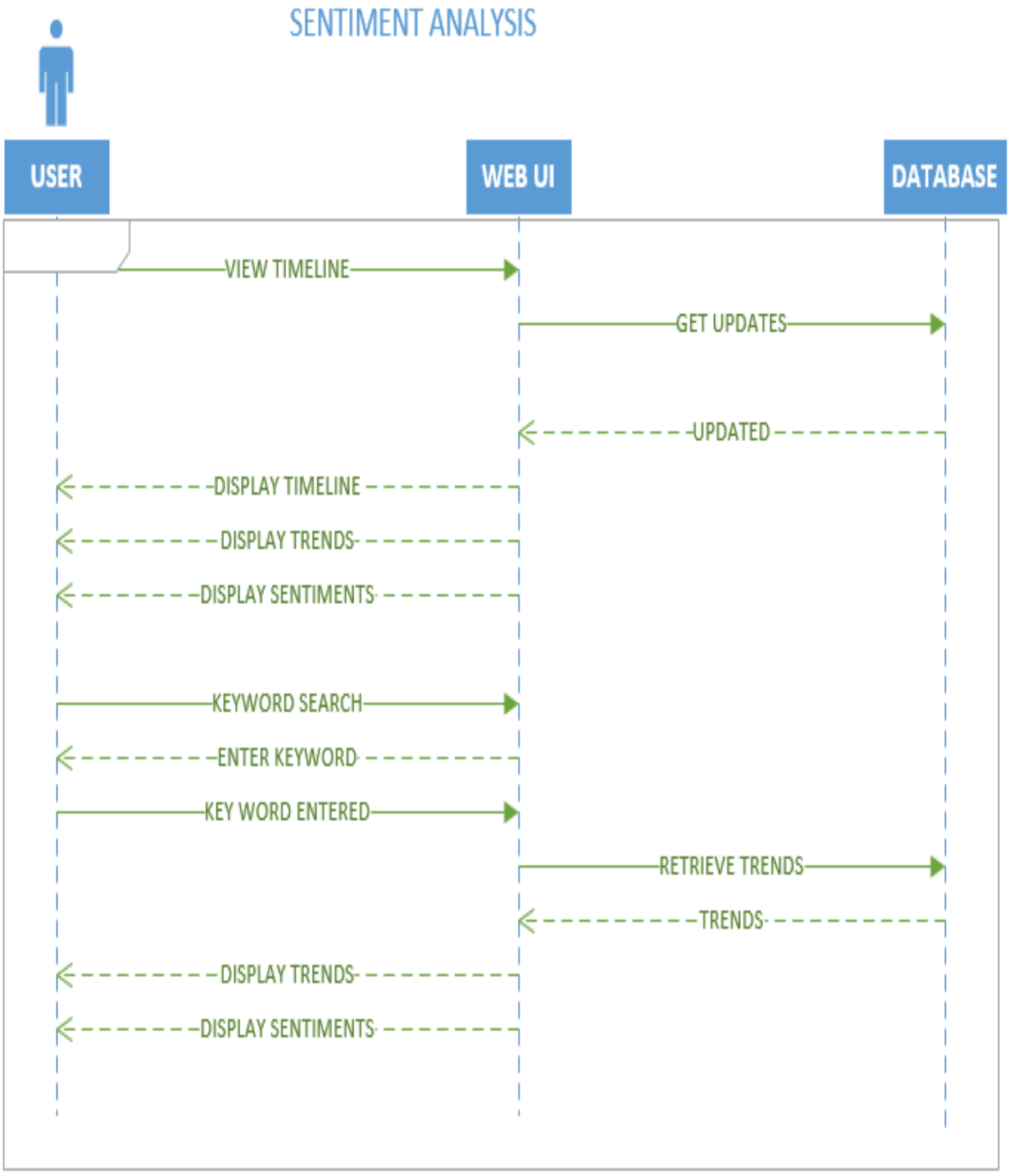

Figure 18-Sequence Diagrams 1: Sentiment Analysis

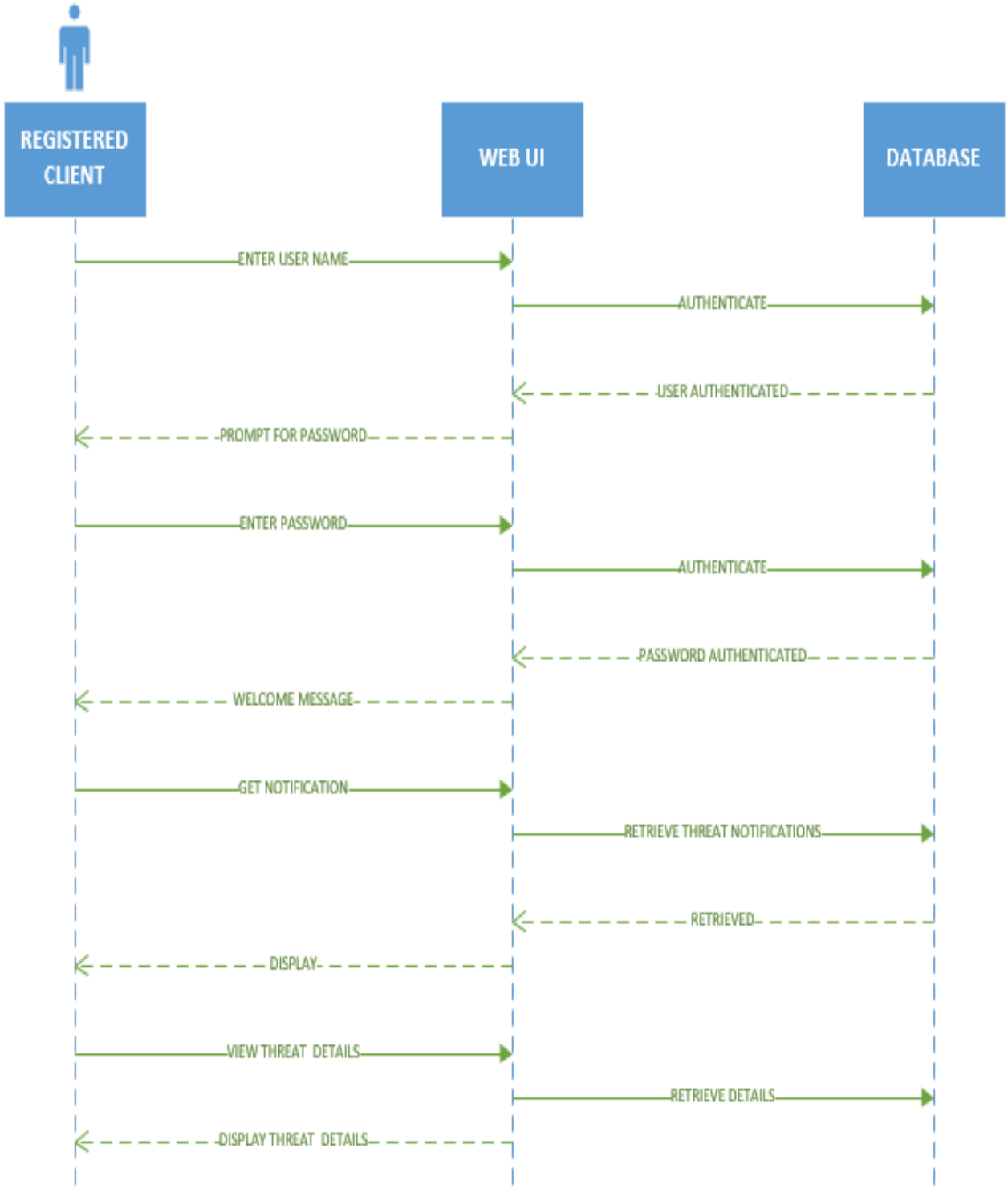

Figure 19-Sequence Diagram2: Threat Detection

4.8.5 Class Diagram

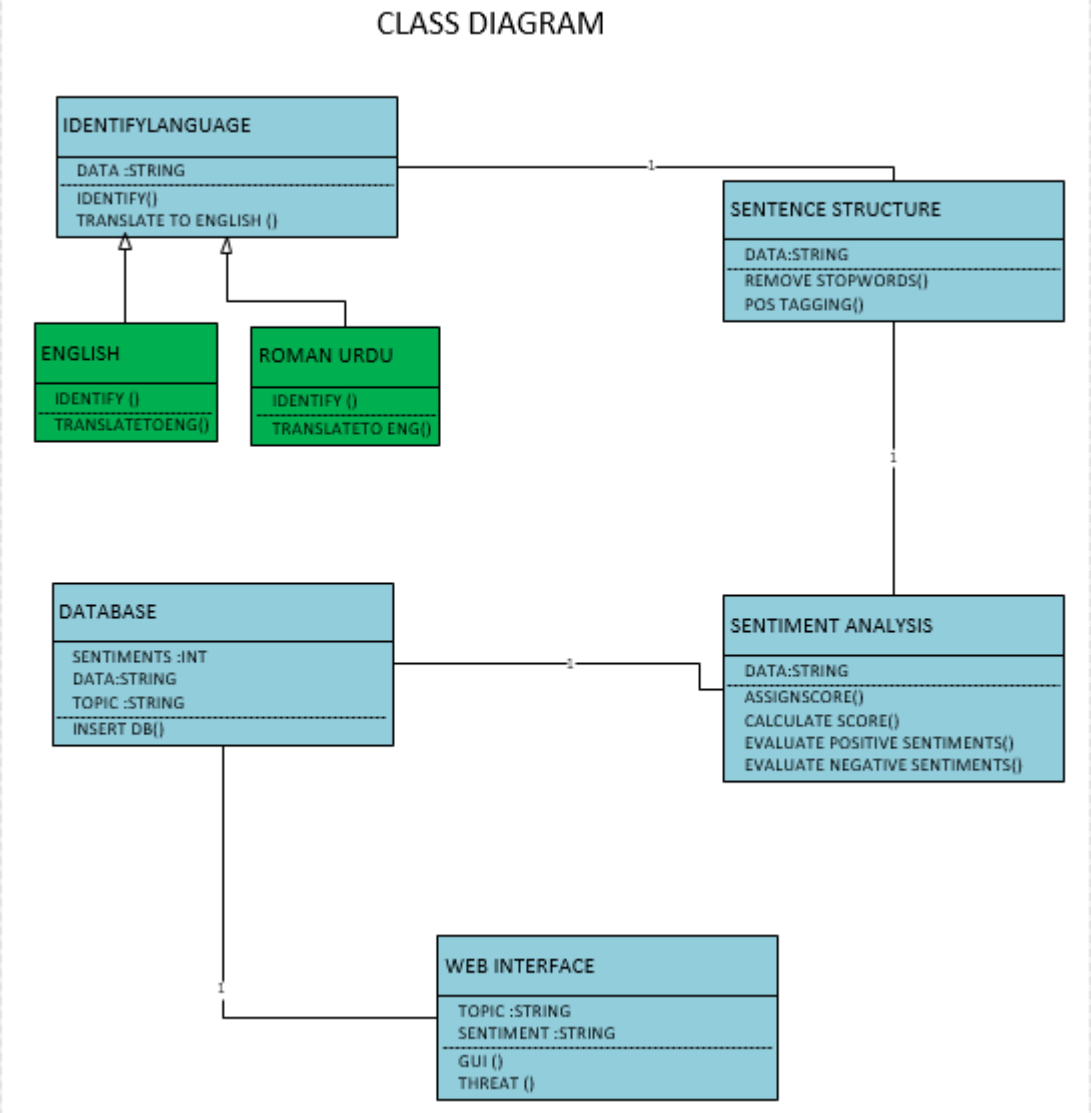

Figure 20-Class Diagram

4.9 Reusability

4.9.1 Reusability in the system

Some off the shelf components, that are available for reuse, are relevant to our project. No module can be entirely replaced by any existing module but smaller components within Text Processing module are being reused without affecting efficiency. These include the dictionaries for Roman Urdu and English. English dictionary is used for identification of user conversations composed entirely in English. English dictionary is also being used for sentence structure extraction. There exist many English dictionaries in forms of software and it is reasonable to use them to save time and put entire focus

on product applications like sentiment analysis and threat detection. Roman Urdu dictionary was rare to find few years ago but there has been a lot of research being done in this subject. It is desired to use someone's research work in our project for identification and translation of Roman Urdu language.

4.9.2 Reusability from the system

There has been a focus on the reusability of components developed in the project as well. Therefore, functionality is kept modular. Functionality for each module is defined with as much clarity as possible. Dependencies between different components of the system have also been demonstrated in this document. Inputs and outputs for all modules are clearly stated and each module properly interfaced so that it can be reused within other projects easily.

4.10 Design Decisions and tradeoffs

The design decision to divide the system up into a four module system was made to promote modularity by keeping the functionality perspective in mind. Modularity provides encapsulation for the important pieces of the system, which allows the developers to make changes with minimized effect on the entire project. For example, if Roman Pashto is to be used instead of Roman Urdu, the only changes made would be in the Text Processing Module. Similarly, adding a new feature such as allowing sentiment analysis for text entered at the web interface would not affect other modules as the new feature will use the interfaces provided by each module for communicating with other modules.

The backend modules, Text Processing and Sentiment Analysis are designed separately to support the modularity in the system. Text processing module makes use of existing software and is designed in a manner so that it can be reused in future projects as well. Sentiment Analysis module provides the core functionality of the system. Reusability factor is kept in mind for designing the core functionality as a separate module.

The database module is kept separate and acts as a bridge between the backend processing and frontend interface. This is desirable as it divides the overall functionality into two halves. Natural Language Processing and Sentiment Analysis are performed at the backend, and results are stored in the database. The web interface simply uses the database for extracting only the desired information. If there is no database between the backend modules and the web interface, we face problems like huge bandwidth requirements and decreased efficiency.

Client Server architecture is used in the development of web interface as there is a centralized database server for communications. Local machines need very little processing capabilities as most of the work is done on the server side. The users only

need to have a stable internet connection. This allows the application to work at good speeds on PCs and laptops as well as mobile devices such as tablets and phones.

4.11 Pseudo Code for Modules

4.11.1 Text Processing Module

```
GetRawUser conversations ()    // Get user conversations  which would be in raw format.

IdentifyLanguage()      // Check whether it is in roman Urdu or English

If(RomanUrdu)
{
Translate(RomanUrdu)    //This function translates into English, takes RomanUrdu as argument.
User conversations user conversations  = ExtractSentenceStructure(User conversations )    //Extract sentence structure, meaning of sentence e g: nouns, verbs and other parts of speech.
SendStructuredUser conversations (user conversations )    // Structured user conversations  passed to module 2
}

if (English)
{

User conversations user conversations  = ExtractSentenceStructure (User conversations )    //Extract sentence structure, meaning of sentence e g: nouns, verbs and other parts of speech.
SendStructuredUser conversations  (user conversations )    // Structured user conversations  passed to sentiment analysis module.
}
```

4.11.2 Sentiment Analysis Module

```
//English structured user conversations

User conversations user conversations  AnalyzeSentiments (user conversations )
{
CalculateScore(user conversations )
AssignScore(user conversations )
Return user conversations ;
}
```

```
Threat threat DetectThreat (user conversations )
{
Extract info from user conversations  with negative sentiment
if(negative sentiment > Threshold)
{
      //Potential Threat
     // Mine the user conversations  deeper to extract information
    Return threat;
}

update (user conversations )        //update database
```

4.11.3 Database Module

```
Create_database ()

Open DBconnection ()

InsertIntoDB (user conversations )

//Create Trigger for finding popular trends from discussion topics, based on the frequency of nouns.
```

4.11.4 Web Interface Module

```
//Update UI (topics) from database.

//Update threat from database

//Data Visualization of user sentiments

//Data Visualization of threat details.
```

5 PROJECT ANALYSIS AND EVALUATION:-

5.1 Test Plan Identifier

This test plan aims to cover UpCycler – A Sentiment Analysis Tool. To make sure that all newly added features of UpCycler are working correctly, this test plan has been created. This test plan shall ensure that all **features testing** of this system is performed accurately. The 1st version of this test plan will cover test scenarios, expected results and acceptance criteria of the product.

5.2 Introduction

UpCycler is a sentiment analysis tool which turns raw data into meaningful data and extract sentiments from it. The test plan is designed to make sure that all features are

working as required. Following are some main features of the UpCycler:

- Sentence Segmentation and POS tagging
- Syntactic Analysis- Parsing
- Identifying Sentiment aspect/Attribute
- Negation handling
- Calculating Polarity
- Sentiment Analysis
- Data Visualization
- Data Integrity check

5.2.1 Objectives

- Identify product information
- Define testing approach
- Target features that have to be tested
- Detail down the test requirements
- Identify testing strategies
- Identify the required resources
- Estimate testing effort
- Schedule testing activities
- Identify risks
- Enlist deliverables

5.2.2 Scope

This document will identify the milestones for test engineers and project managers. It will provide the framework for high level component testing of the UPCYCLER. The testing efforts will be planned for component, integration and System level. As the system is a large set of modules so it is necessary to prioritize the module and testing shall take place accordingly.

5.3 Test Items

This section will provide test items for unit testing. Black box testing will be performed on Subsystem interface. External interface for the following browsers will be tested:

- Firefox
- Opera
- Google chrome
- Browser for Mobile Phone
- Internet explorer

5.3.1 Sub-systems for Test

- Manage timeline
- Manage notifications

- Manage visualization

5.3.1.1 Performance tests

Response Time for:

- o Login
- o Updating Timeline
- o Visualization
- o Requested services

5.3.1.2 Functionality test:

- o Login as authenticated user
- o Login as guest
- o Search topics
- o view Notification details
- o Data visualization
- o Maintain Timeline
- o Maintain Dashboard
- o Logout

5.3 Features to Be Tested

5.4.1 Functionality Tests

Test Case Name	Login Authenticated User
Test Case Number	1
Description	User should have been registered
Preconditions	User should have been registered
Input	Email and Password
Steps	After getting the User email and password it will check for its authenticity and if the user is authentic he/she will be logged in.
Expected output	Three Tests for login authenticated user
Results	Three Tests in which authenticated user was logged in.

Test Case Name	Login Guest
Test Case Number	2
Description	No authenticated required. Only Login as a guest
Preconditions	Application should be open
Input	Click the Login as guest button
Steps	After clicking on Login in as guest button, home page of application will be displayed.
Expected output	Three Tests for login guest user
Results	Three Tests in which guest user was logged in.

Test Case Name	Search Topics
Test Case Number	3
Description	User enters a keyword to search for the relevant topic in order to find out user sentiments about it
Preconditions	User should be logged in.
Input	Enters a key word.
Steps	First user enters a key word then clicks on the search button the relevant results are displayed.
Expected output	Three Tests for search using keyword.
Results	Three Tests in which search was done using keyword.

	View Notification Details

Test Case Name	
Test Case Number	4
Description	Only authenticated user can view user conversations that Contains potential threat and can see details about it.
Preconditions	User should log in as authenticated user.
Input	Click view Notification Button
Steps	Authenticated user clicks on view notification button and relevant notifications are displayed.
Expected output	Three Tests for view notification details
Results	Three Tests in which notification details were viewed.

Test Case Name	View Visualization Details
Test Case Number	5
Description	User Sentiments in the form of bar-graphs, Donut charts etc. will be displayed
Preconditions	Access the web application
Input	Click to view data visualization
Steps	Visit web application Click on Topics
Expected output	Three Tests for data visualization.
Results	Three Tests in which data visualization was viewed.

Test Case Name	Maintain Timeline
Test Case Number	6
Description	Timeline would display all the topics uses are talking about.

Preconditions	User should be logged in.
Input	Topics and trends
Steps	Sentiments and opinions about topics are displayed
Expected output	Three Tests for Maintain timeline.
Results	Three Tests in which timeline was maintained

Test Case Name	Maintain Dashboard
Test Case Number	7
Description	Contain topics about which people are most interested In.
Preconditions	Database is connected.
Input	Topics and trends
Steps	Sentiments and opinions about topics in which user is most interested are displayed
Expected output	Three Tests for maintain dashboard
Results	Three Tests in which dashboard was maintained

Test Case Name	Logout
Test Case Number	8
Description	User would exit from the current session
Preconditions	User should be logged in.
Input	Click on the logout button
Steps	User clicks on the logout button and exists current session.

Expected output	Three Tests for logout.
Results	Three Tests in which timeline was maintained

5.4.1 Quality Tests

Features	Priority	Description
Notification	1	Ensure the receiver
Authentication	1	For confidentiality
Restrictions	1	Filtering is needed to make sure data is visible to required viewer

5.4.3 Technical Tests

Features	Priority	Description
Security of Data	1	Access to database should be restrictive and under the management control

5.4 Features Not To Be Tested

Features	Description
Web Security	Out of scope
Networking performance	Out of scope

5.5 Approach

5.5.1 Testing Process

The process that will be followed in testing is as follow

5.5.2 Unit Testing

Logic error, syntax error and other language specific errors shall be identified at source code level. Unit test cases shall to ensure the validity of modules correctness.
This particular test will be done by developers.

5.5.3 White Box Testing

UI is not considered in tests during white box testing. At code level program inputs and output are tested and validated against specification. This type of testing focuses on structure of the code and functions of the programs are ignored. Modules which will be tested at code level are:

- Data preprocessing
- Syntactic Analysis
- Extracting Sentiment Aspect
- Negation Handling
- Named Entity Recognition
- Calculating Polarity
- Evaluating Sentiments

White box testing will be achieved through 2 different test coverage techniques:

- Branch coverage testing
- Multiple conditional coverage testing

5.5.4 Black Box Testing

Black box testing involves verifying results by running through all possible inputs and the output is verified. It is perceived that an end user will enter all possible inputs. Tester shall perform:

- Equivalence class partitioning and
- Boundary value analysis

5.5.5 Integration Testing

Integration testing shall be performed to ensure that all dependent components run

nicely together. And none of component is the cause of performance degradation for other component.

5.5.6 System Testing

To verify the agreed requirements system testing is performed. Following objectives is to be met to ensure that defects found during testing are corrected properly:

- Functional requirements are implemented
- System internal interfaces are implemented
- System external interface are implemented

5.5.7 Load Testing

Tester shall perform the Load testing using related tool to verify that sufficient user can be stay online at single instance. This test will ensure loading quality of the Up Cycler.

5.5.8 UI testing

This testing technique shall evaluate the user interaction with the system. The objectives include:

- Easy navigation through functions
- Objects in UI are according to industry standards
- Nothing in UI bothers the user

5.6 Environmental Needs

5.6.1 Hardware

Server with at least 2 MB dedicated internet connection

5.6.2 Software

Following dependency should be met:

Microsoft IIS Express server
Microsoft SQL Server 2014
Microsoft Visual Studio 2012
Python IDLE
Net beans 8.2

5 FUTURE WORK

Since sentiment analysis is such a popular topic nowadays, the project ‘UPCYCLER’ can be extended to try out various methods for extraction of topics and associated sentiments of people. A lot of research work is being done to improve the accuracy of sentiments. In addition to that, instead of one general topic, it is proposed that different

aspects of a product, topic or event are analyzed for collecting user sentiments as it adds huge value to the original task. Future improvements in the model may attract companies, politicians, and other potential clients who value the sentiments of general public.

7. CONCLUSION

UPCYCLER is a complete package with a lexicon based sentiment analysis model working at the backend and an eye-catching frontend web application. It can be used by general public to review the overall sentiments about trending topics amongst themselves. Users can get a general idea about the public sentiments. Visual representations of sentiment analysis add value to the overall project. The project contributes to the overall society, and can be regarded as an academic success.

BIBLIOGRAPHY

- Schlect, G., Sacksteder, R., Bloch, E. & Trey, P. Social Mood Swing, Retrieved from: http://www2.cs.uidaho.edu/~cs480b/documents/TEAM_GREP_SRS.pdf.
- Bing, L. Sentiment Analysis and Opinion Mining. Retrieved from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.244.9480&rep=rep1&type=pdf
- Pang, B. & Lee, L. Opinion Mining and Sentiment Analysis. Retrieved from: http://www.cse.iitb.ac.in/~pb/cs626-449-2009/prev-years-other-things-nlp/sentiment-analysis-opinion-mining-pang-lee-omsa-published.pdf

- Ambler, S. W. (2014). UML 2 Use Case Diagrams: An Agile Introduction. *Agile Modeling.* Retrieved on 18th December, 2015 Retrieved from: http://www.agilemodeling.com/artifacts/useCaseDiagram.htm

- Ambler, S. W. (2014).). UML 2 Activity Diagrams: An Agile Introduction. *Agile Modeling.* Retrieved on 18th December, 2015 Retrieved from: http://agilemodeling.com/artifacts/activityDiagram.htm

- Sparx System (2015). UML 2 Activity Diagram. *Sparx Systems.* Retrieved on 19th December, 2015. Retrieved from: http://www.sparxsystems.com/resources/uml2_tutorial/uml2_activitydiagram.html

- Bell, D. (2003). UML basics: An introduction to the Unified Modeling Language. *IBM Global Services.* Retrieved on 19th December, 2015. Retrieved from: http://www.nyu.edu/classes/jcf/g22.2440-001_sp06/handouts/UMLBasics.pdf
- Booch, G., Rumbaugh, J., & Jacobso, I. (1998). *The Unified Modeling Language User Guide.* Boston, MA: Addison Weslay.
- MSDN (2015). UML Class Diagrams: Reference. Retrieved on 20th December, 2015. Retrieved from: https://msdn.microsoft.com/en-us/library/dd409437.aspx