ÇANKAYA UNIVERSITY

FACULTY OF ENGINEERING

COMPUTER ENGINEERING DEPARTMENT

Project Report

Version 1

CENG 407

SMS DETECTION MOBILE APPLICATION

Sarper SİNANOĞLU- 201511052

Muazzez AYPEK- 201411009

Emre GEDİKOĞLU- 201511024

Berkcan ÇELİK- 201411015

Advisor: Dr. Instructor *Roya CHOUPANI*

Co-Advisor: Dr. Instructor *Faris Serdar TAŞEL*

November 27st , 2019

Table of Contents

Abstract

Short Message Service (SMS) is one of the basic, yet most significant means of communication in the modern world. Similar to all means of communication, the world of SMS communication is also prone to controversy. This brings us to the topic of spam SMS detection. Although the use of real-time messaging applications has a tendency to surpass the preference of SMS usage nowadays, communication via SMS is still widely used throughout the world. Thus, a necessity of sorting out and filtering different varieties of SMSs is born. In this report, we will address SMSs as either spam or legitimate. Although many definitions are possible, we can simply refer to a spam SMS as an SMS that contains unnecessary/unwanted content or puts the user at risk. The risks include loss of data, invasion of privacy and the unauthorized access of personal or confidential information. To choose between a legit SMS and an SMS as described above, spam SMS detection has proven to be vital. As a result of our research, we came to a realization that there does not exist such a mobile application which is capable of detecting whether a certain SMS is spam or not. There does exit previous work on how to detect and handle spam SMS. However, we intend to improve and contribute to this previous work by putting forth a mobile application which will assist mobile users in sorting out and filtering their SMS inboxes. Within our report, we will explain the necessary methods and algorithms for the creation of our spam SMS detection mobile application.

Keywords: Spam SMS Detection, Machine Learning Classifiers, Deep Learning Techniques, Data Analysis, Data Cleaning, Pre-processing.

1. Introduction

With billions of mobile phone users sending trillions of text messages (SMSs) daily, SMS communication is widely utilized. The number of spam SMSs sent worldwide has shown an increase with respect to the increase in SMS utilization. So naturally, the increase in the number of spam SMSs poses a great risk for mobile phone users engaging in SMS traffic. As mentioned in the abstract, this risk can be categorized into loss of data, invasion of privacy and the unauthorized access of personal/confidential information. Data such as lists of contacts, image files, credit card numbers and passwords are vulnerable to attacks from malware and key-loggers that have hidden their true intentions behind the contents of spam SMSs. In addition to problems surrounding the security and privacy of mobile phone users that are actively taking place inside their respective SMS traffic, spam SMSs simply cause irrelevancy inside the SMS inboxes of mobile phone users by crowding the inbox with unnecessary and unwanted text messages. Similar to the e-mail environment, users do not want to see SMSs that are not of their interest or potential liking. The goal of this project is to serve as a defense wall against the security and privacy issues of the SMS environment and to provide a much more relevant and understandable SMS inbox to mobile phone users. To serve as a solid basis for our project, we conducted a thorough research evolving around the previous work related to our topic of spam SMS detection. We observed that many of the previous work surrounding the spam SMS issue was of research level and did not transform into a user-level product. We came upon various methods and algorithms related with this area of work. Mentioned here as briefly as possible, these methods/algorithms are Naive Bayes, Support Vector Machine, Convolutional Neural Network, Logistic Regression, Long Short Term Memory, Gated Recurrent Unit and Apriori. This research level previous work will be explained in detail, below in the Related Work section of this report. Aside from many research level previous work that we have encountered, we are also aware of a few user-level spam SMS detection products. However, we find them to be insufficient in terms of the ability to distinguish SMSs as either spam or legitimate in a satisfactory manner. This is mainly because they perform filtering and detection just by telephone number or keywords, which are manually entered by the user. These end products did not seem to be efficiently utilizing the concepts of machine learning and deep learning. These concepts are primarily what we want to add to the existing solutions for the spam SMS detection issue. Here in the Introduction section of our report, we tried to give an overview of our research in the spam SMS detection area and aimed to simply put forth what our goal is and how we intend to fulfill it.

Throughout the rest of this report, we will give the details of our research and pave our way in terms of developing an efficient spam SMS detection mobile application.

2. Literature Search

2.1 Related Work

2.1.1 A General Overview on the Necessity of Spam SMS Detection

As mentioned in the Introduction section of our report, the number of spam SMSs has shown a significant increase as the number of SMSs sent daily throughout the world has increased. As an example, we came upon a research that stated that the annual worldwide SMS traffic had risen to over 6.9 trillion all over the world at the end of 2010. This rise in the SMS traffic created an opportunity for spammers to carry out their disturbing work, this being so unfortunate for mobile phone users worldwide. [1] Furthermore, research has it that 30% of Americans received and average of 8.4 spam SMSs monthly in 2015. [4] Previously, we gave a brief explanation as to what spam SMS detection is and why it has proven to be necessary. Within the contents of the articles which we have read through for our research, the reasons behind the need for detecting spam SMSs were taken into hand in terms of different aspects. We can examine these aspects as privacy/security issues and relevancy issues. Mobile phone users obviously want to maintain their privacy, send and receive SMSs in a secure fashion and of course, receive SMSs that are relevant with respect to their interests.

If we concentrate around the phrase “relevancy with respect to one’s interests”, we see that there are many kinds of spam SMSs that violate this will of mobile phone users. So, what are the kinds of spam SMSs that are not of most mobile phone users’ interests? Credit proposals of banks, promotion and discount notifications of various brands, misleading lottery win announcements, general advertisement messages of all kinds of companies and most importantly fraud. [2] Just for clarification, we would like to draw attention to one point. In some instances, we tend to give out our telephone numbers during surveys, at the cash registers of stores while we are paying, to sign up for websites, etc…In such situations, we are purposely handing out our telephone numbers in order to be kept up to date by the companies that we have filled out surveys for, stores that we have bought from and the websites we have signed up for. Thus, SMSs received from these kinds of senders cannot be thought of as spam SMSs, and need to be especially dealt with during the spam SMS detection process.

Now that we have discussed the relevancy issues that comes along with the spam SMS detection concept, we need to zero in on the privacy and security threats that spammers bring in to the equation. Just like e-mail spammers, SMS spammers also pose potential of serious danger for mobile phone users. Generally speaking, SMS spammers are seeking to extract personal/confidential information from the smart phones of mobile phone users. They aim to exploit any kind of data leakage that they are able to cause and have a reach at all kinds of unauthorized data that they are able to hack into. When we say personal/confidential information or unauthorized data, we can think of many aspects. To suggest a few: list of contacts, image files, credit card numbers of individuals and passwords. [3] To access these types of data, spam SMS senders have a tendency to install malicious software into the phones of mobile phone users via the harmful text message they send. Scam, fraud, man in the middle attacks and viruses are also possible threats imposed by spam SMSs. [4]

In addition to the relevancy and privacy/security issues resulting from the sending of spam SMSs, we also came upon a few more problems caused by such text messages. First of all, the importance of time for us all is obviously not to be discussed. Spam SMSs simply steal our time if we take into consideration the time we spend to distinguish between which SMS is spam and which SMS is not. This may not seem time consuming at first, but when thought of on a weekly or monthly basis, this causes an obvious loss of time for an average mobile phone user. Secondly, to the surprise of many, spam SMSs even have the potential to harm human health. As an example, we can mention the harmful health product (weight loss products for instance) promotions dealt out via text messages. [5]

This section of our report concentrated on the various issues and threats that revolve around spammers and spam SMSs. After carrying out our research and encountering the information, regarding spam SMSs, that we have shared above, we have decided to develop a spam SMS detection mobile application to contribute to the solution of such a significant problem in the telecommunication world.

2.1.2 Algorithms and Methodologies used in Previous Work

Now that we have covered what spam SMS detection is and why it is necessary, we have come to the section of our report in which we will talk about the algorithms and methodologies used in the previous work for spam SMS detection, that we have encountered throughout the course of our research. The algorithms and methodologies that we will concentrate on are text preprocessing text preprocessing, Tokenization, feature extraction (Bag of Words), Support

Vector Machine (SVM), Naive Bayes, Convolutional Neural Network, Logistic Regression, Random Forest.

Before mentioning the algorithms and methodologies, we would like to briefly talk about text preprocessing and feature extraction. Text preprocessing is basically getting the text ready for placing it into the machine learning algorithms and methodologies as input. Since we will be working with text messages (SMSs), our data is composed of text and text only. Getting rid of non-alphabetic characters and transforming all characters into lowercase are two examples of text preprocessing.

On the other hand, we can define feature extraction as representing documents as numerical vectors with the same size. Some features that can be utilized are nouns, adjectives, verbs, stop words, one letter words, etc…[7]For feature extraction, we plan on using the Bag of Words method. Bag of Words is a way of modeling text documents, in order to make them compatible with machine learning techniques. Text documents are often messy, as opposed to how machine learning techniques would prefer them to be. Well defined inputs and outputs of fixed length are what we need, to work with machine learning algorithms. [23] This is possible with feature extraction or feature encoding, as mentioned above. A Bag of Words representation shows the occurrence of individual words in a given text document. It involves two main aspects. These are: A vocabulary that includes known words and a certain measure of the presence of these known words. Once these aspects are satisfied, documents are transformed into vectors, as is necessary for feature extraction. [24]

In cooperation with the Bag of Words method, we will need to perform tokenization. Tokenization can be defined as: Dividing a given textual data into meaningful pieces called "tokens" in order for them to be processed via machine learning algorithms and techniques. Tokens are grouped into types according to their sequence of characters. [8] Below, Figure 1 shows an example for tokenization in Python. As mentioned above, textual data needs to be preprocessed before being used as input for machine learning algorithms and techniques. After preprocessing, this data needs to be tokenized and converted into numerical vectors of equal sizes. Thus, we too will apply these techniques to our data before utilizing the algorithms and methodologies that we will describe below.

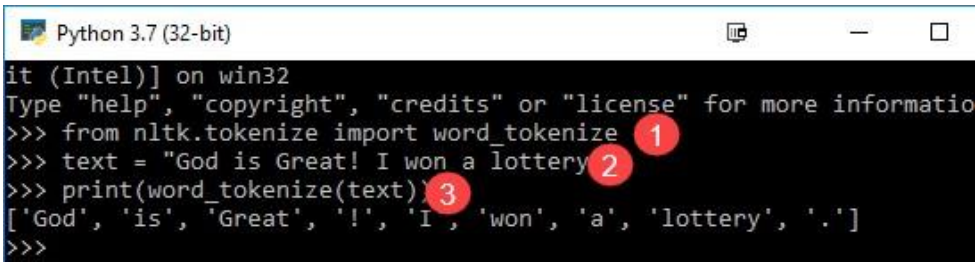

Figure 1: An example of tokenization in Python. [9]

We will begin with Support Vector Machine (SVM). SVM is known as a "linear classifier" in which a training set is gathered from the data set. SVM attempts to find a hyper plane that acts as a classifying boundary that separates the training set into two different classes. The data points (in both classes) which are located as close as possible to the hyper plane are known as "support vectors". With SVM, comes a term known as a "margin". The margin is the distance between the support vectors. The preferred hyper plane is one that produces the greatest margin. [6] A graphical representation for Support Vector Machine (SVM):

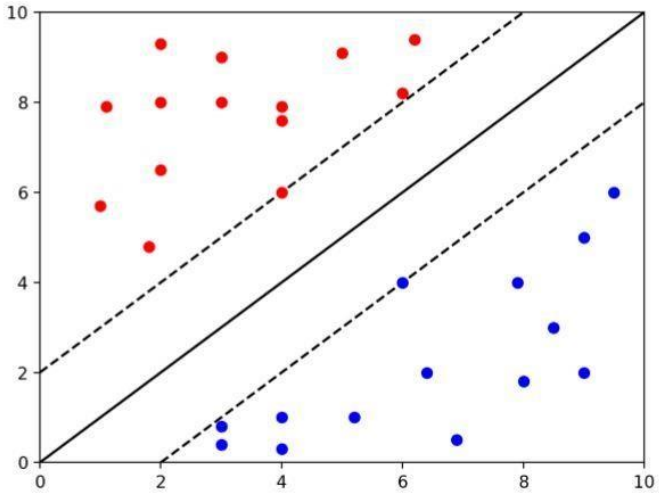

Figure 2: A graphical representation of Support Vector Machine where the dashed lines represent the support vectors and the main diagonal represents the hyper plane. [6]

A few advantages of using the Support Vector Machine Algorithm are:

- The use of support vectors enables for a more efficient use of the memory.
- There are various kernel functions that can be used to form the hyperplace. This provides flexibility. [17]
- The major advantage is that SVM transforms the classification problem into a quadratic optimization problem. This way, the number of processes in the machine learning phase can be decremented. As a result, faster solutions can be produced in comparison to other techniques and algorithms. [18]

The next algorithm that we examined during our research was the Naïve Bayes algorithm. Naïve Bayes is a probabilistic classifier that works independent assumptions. It is based on the Bayes theorem and treats each word in a given SMS independently. The frequency of the words in the training set must be counted and these occurring words are treated independent from each other. Basically speaking, the algorithm depends on the presence or absence of words to mark a certain SMS as either spam or legitimate. Above, we mentioned the term feature extraction in which we are going to find features of the tokens that we acquire after the tokenization process. The Naïve Bayes algorithm assumes that all features are independent, just like the words themselves. Factors that make the Naive Bayes algorithm appealing are:

- The independence of words in a given textual data.
- Locations of words are not dependent on one another.
- The assumption of the independence of the features of data elements with respect to each other (this is what makes the algorithm “naive”.
- Works cooperatively with the Bag of Words method.
- Easy to train.
- Has a fast way of classifying data objects.[11]

Below, Figure 3 shows an implementation of the Naive Bayes algorithm:

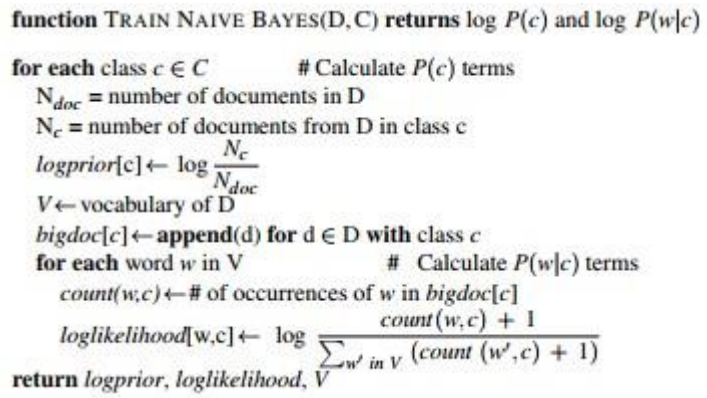

Figure 3: An implementation of the Naive Bayes algorithm in Python. [14]

As a third algorithm, we came upon Convolutional Neural Network (CNN). CNN is known as a deep learning classifier. We previously talked about the need for feature extraction when using machine learning algorithms. These algorithms depend on manual feature extraction, which so naturally depends on previous domain knowledge. Deep learning enables for a classifier that does not need manual feature extraction because it automatically identifies the hidden features from the data set. [7]

The algorithm works in three phases, these being: word matrix formation, hidden feature identification and classification. Below, we have shared Figure 4, which shows the Word matrix formation for the Convolutional Neural Network algorithm.

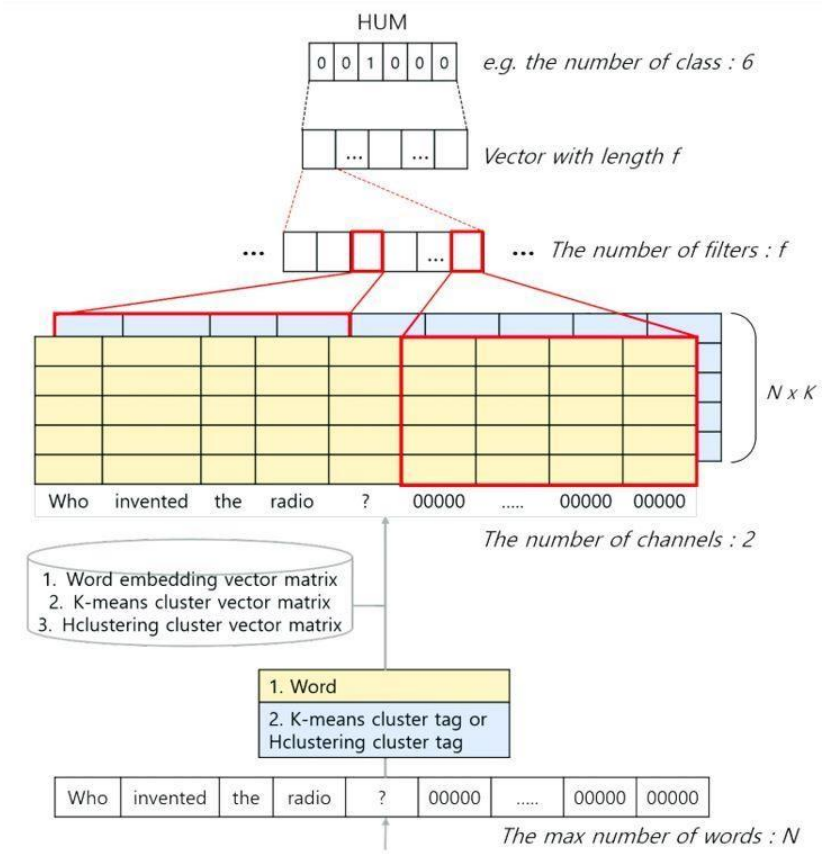

Figure 4: The word matrix formation phase of a CNN based sentence classification with semantic features using word clustering. [13]

After this phase, the newly formed word matrices are used to identify the hidden features of the data set. Once the features are extracted, the Convolutional Neural Network algorithm will be able to classify the data objects into the previously defined classes. [12]

If we have to mention a few advantages of using the Convolutional Neural Network, we could say that:

- CNNs have the ability to capture and learn features at various levels, just like the human brain.
- In terms of memory and complexity, CNNs are more efficient when compared to NNs (Neural Networks), because they enable weight sharing.
- As mentioned above, CNNs are significantly successful feature extractors. Thus, they provide the benefit of not having to manually extract features from the data. Instead, hidden features of the data set are automatically identified. [10]

One other algorithm that we encountered was the Logistic Regression algorithm. Logistic regression, like the Naive Bayes algorithm, is also a machine learning algorithm. It provides an analysis based on prediction and is used for classification purposes (in our case, text classification). Like Naive Bayes, Logistic Regression is also based on probabalitity, thus making it a probabilistic classifier.

Logistic Regression shows similarity to Linear Regression but utilizes a much more complex cost function, called the sigmoid function (also known as the logistic function). The cost function is necessary because it represents an objective of optimization. The minimization of the cost function enables for the formation of a model with minimum error. To minimize the cost value, Gradient Descent is used. Due to the hypothesis of Logistic Regression, the cost function is limited between 0 and 1. Just as in other machine learning algorithms, the sigmoid function in Logistic Regression maps a given prediction to a probability value between 0 and 1. Thus, we obtain a probabilistic classifier that takes inputs that are previously passed through a prediction function and returns probability score between 0 and 1 as an output. Below, Figure 5 shows a graphical representation of the sigmoid function used in the Logistic Regression algorithm. [15]

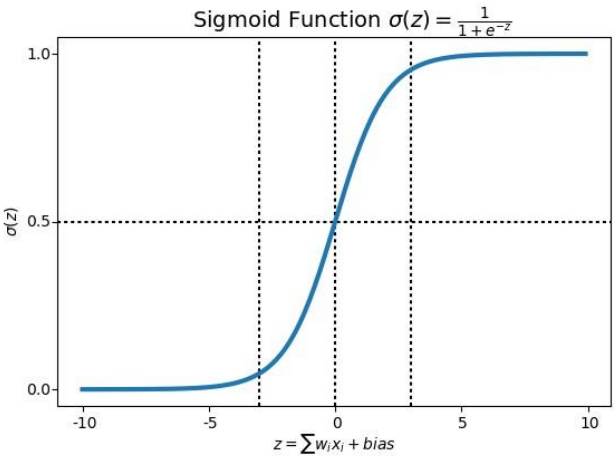

Figure 5: A graphical representation of the sigmoid function. [15]

Before moving on, we would like to mention a few advantages of using the Logistic Regression algorithm. These are:

- It is robust in the sense that independent variables do not have to be normally distributed.
- It has the ability to handle nonlinear cases.

- It provides a maximum likelihood estimation by transforming dependent variables into logit variables (with respect to the independent variables). [16]

The final algorithm that we concentrated on was the Random Forest algorithm. Just like the name states, the algorithm produces a wide range of decision trees that form a "random forest". Each decision tree in the forest gives a slightly different prediction than the other trees. By combining all of the predictions that have been acquired from the decision trees in the forest, the algorithm enables for a significant improvement in the final overall prediction. To put it more simply, the combination of many decision trees produces a better result than relying on just one tree. To mention some advantages of the Random Forest Algorithm, we can say: [19]

- The Random Forest algorithm relieves us of the overfitting problem that is very common in decision trees. [20]
- It is robust to outliers and uses binning to handle them.
- When compared to regular decision trees, the Random Forest Algorithm is faster to train. Additionally, prediction speed is also faster than decision trees, given that we are able to save previously generated trees for later use in the future. [21]

A graphical representation for the Random Forest algorithm is given below in Figure 6:

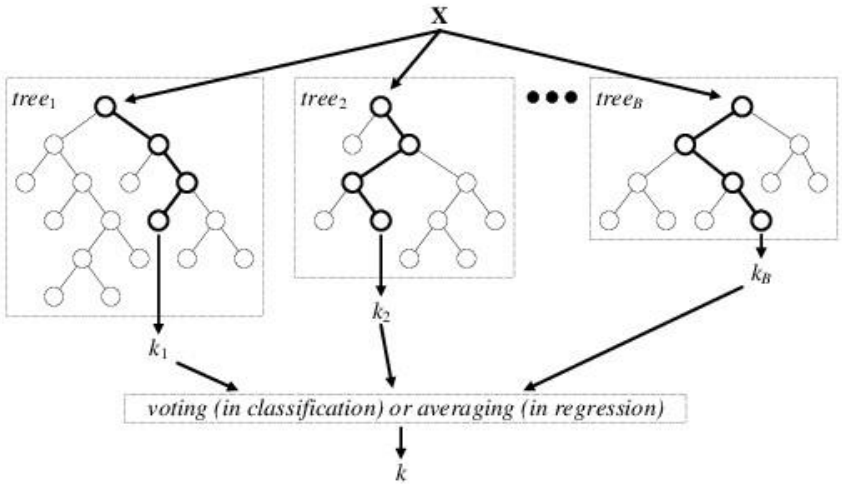

Figure 6: The graphical representation of the application of the Random Forest algorithm for classification and regression purposes. [22]

In this section of our report, we aimed to give a thorough explanation of our research on the algorithms and methodologies used in the previous work done on spam SMS detection. After examining the various algorithms and techniques, we were able to gain an insight on how other researchers and project teams approached the topic of spam SMS detection and filtration. We concentrated on the working principles and implementations of these algorithms and methodologies and had a chance to compare and contrast between them. At the end of our research, we were able to come up with an idea as to how we are going to approach the topic and carry on with our project. We will talk about the results we acquired and the algorithm we decided on using, in the next (Results of our Research) section of our report.

2.1.3 Language and Dataset

Our spam SMS detection mobile application will be developed for the English language. There are two reasons behind our choice of English. These are:

- We would like to appeal to the widest range of users possible. Given that English is the *lingua franca* of our modern day, basing our application on the English language will allow us to gain the maximum number of users.
- Datasets that contain ham-spam (a “ham” SMS refers to any SMS that is legitimate) SMSs are most commonly encountered for the English language. Altough datasets for other languages are also present, they are not as detailed as those in English. Once again, this encourages us to develop our application for the English language.

As our dataset, we will be using the SMS Spam Collection Dataset from the UCI Machine Learning Repository. [26]

2.1.4 Results of our Research

With our research coming to an end, we were finally able to decide on which machine learning algorithm suited the work area of spam SMS detection the best. While making our decision, we also gave a great deal of importance to the compatibility of the algorithm of our choice, with the feature extraction model (Bag of Words Model) that we have already decided on utilizing. With that said, we concluded that we will carry on with our project with the Naïve Bayes classifier. Below is a figure that shows a comparison between the accuracies of Naïve Bayes, Random Forest and Logistic Regression in means of SMS spam detection and filtration (all of which, we included in our report):

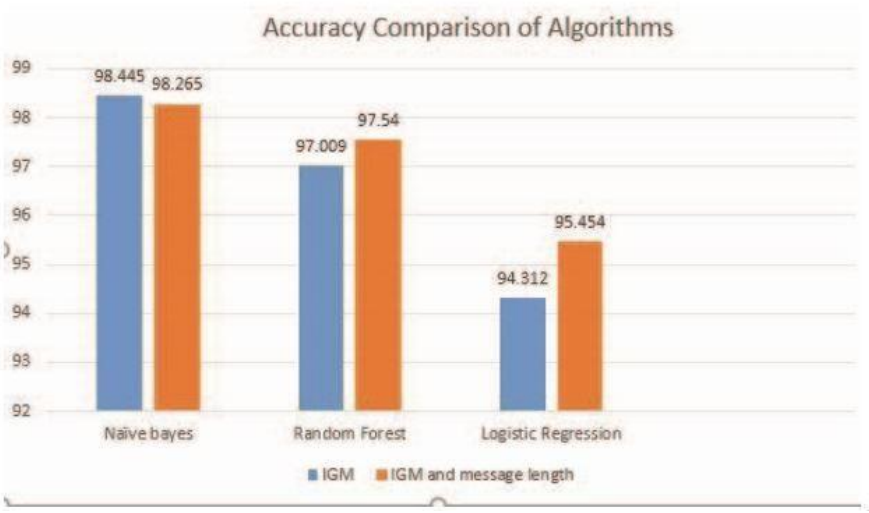

Figure 7: A comparison of the accuracies of Naive Bayes, Random Forest and Logistic Regression algorithms in terms of SMS spam detection. [25]

As can be seen in Figure 7, the accuracies of the three algorithms are analyzed in two cases: one in which only the information gain matrices are considered, and the other in which both the information gain matrices and the message lengths are taken into account. In both cases, the accuracy of the Naïve Bayes algorithm surpasses the accuracies of the Random Forest and the Logistic Regression algorithms.

Aside from the accuracy advantage of the Naïve Bayes algorithm (which is, of course, a significant determinant), the cooperation of the Naïve Bayes algorithm with the Bag of Words feature extraction model also acts as a deciding factor behind our choice of this particular algorithm. The figure below shows the cooperation of the Naïve Bayes algorithm with the Bag of Words, Bigram and Word2Vec feature extraction models.

Naive Bayes					
Bag of Words		Bigram		Word2Vec	
Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
0.978	**0.918**	**0.974**	**0.909**	**0.961**	**0.866**

Naive Bayes does not have any parameters that can be changed. But quickly it can be concluded that Bag of Words outperform Bigram and Word2Vec, when measuring accuracy and F1-score.

Figure 8: A comparison of the cooperation of the Naive Bayes algorithm with the Bag of Words, Bigram and Word2Vec models in terms of accuracy and F1-score. [6]

Bigram and Word2Vec are two other feature extraction models. However, they are shown here merely for comparison purposes. As can be seen in Figure 8, the Naïve Bayes algorithm's cooperation with the Bag of Words feature extraction model is better in means of accuracy and F1-score in comparison to the Bigram and Word2Vec models.

In addition to its compatibility to the Bag of Words method and the fact that it surpasses the other machine learning algorithms in terms of accuracy, the Naïve Bayes algorithm provides various other advantages that we have covered in the previous sections of our report.

With these said, we have formed a data classification model for our spam SMS detection mobile application, which we will go through step by step. Our model includes text preprocessing, text tokenization, feature extraction/vectorization and training/testing the Naive Bayes machine learning classifier. We will take a proportion of 70:30 for the training:testing phase. To better represent our classification model, we have created a graphical representation of it. This graphical representation is given below:

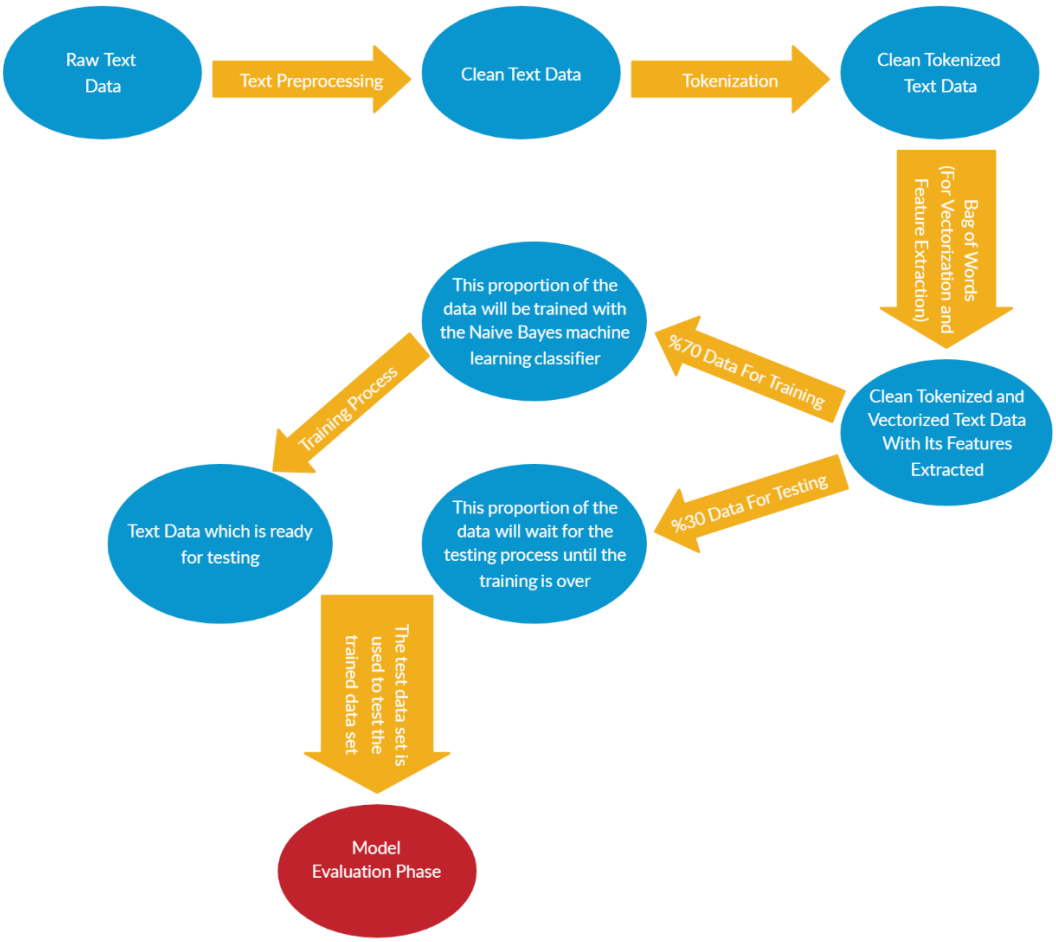

Figure 9: A graphical representation of our text classification model.

2.2.Conclusion

In this report, we attempted to define the concept of spam SMS detection, as well as explain the necessity towards such a topic. After conducting our research on the previous work based on spam SMS detection, we were able to gather significant information regarding the algorithms and methodologies that other researchers and project teams put to use in their different approachesto the subject. This information that we gathered enabled us to compare and contrast between various techniques and create our own approach to the work area of spam SMS detection. We included all of our thought steps in this report and formed the basis for our spam SMS detection mobile application.

To conclude, we would like to restate the importance of spam SMS detection for the telecommunication world in the modern day. With the extensive use of text messaging, mobile phone users are constantly face to face with the threats imposed onto them by spammers, who are on the quest to confiscate their personal/confidential information. In addition to them imposing threats, spam SMSs are simply a waste of time and result in SMS inboxes that are irrelevant with respect to the interests of mobile phone users. With these said, we would like to sum up our report by emphasizing on our goal, which is: to develop a spam SMS detection mobile application for the iOS and Android mobile operating systems, that will provide the users a much more convenient text messaging service that is user-friendly, safe, tidy and relative to their needs andinterest.

3. Software Requirements Specification

3.1. Introduction

3.1.1.Purpose

The purpose of our application is to provide a safe and comfortable text messaging experience to our users by detecting and filtering the annoying and threatening spam SMSs that they may receive from unwanted senders. We intend on efficiently using text classification and machine learning techniques to enable our application to perform in a way that promises an accurate detection and filtration of spam SMSs. We have already conducted a thorough research on the previous work done on the spam SMS detection topic and plan on utilizing our deductions from our research as a contribution to the work we are going to carry out in the Software Requirements Specification phase of our project.

3.1.2. Scope of the Project

The goal of our project is to contribute to the world of telecommunication by developing a spam SMS detection mobile application which mobile phone users will be able to benefit from in means of a much safer and convenient text messaging experience, along with an SMS inbox that is more relevant with respect to their needs and interests. We found out during our previous research that the daily SMS traffic has risen to over 6.9 trillion text messages. This situation, unfortunately, suits the wrongful intentions of SMS spammers. These wrongful intentions can be listed as below:

- The confiscation of the personal/confidential information of mobile phone users is certainly what SMS spammers are mostly eager about. This information includes contact lists, credit card numbers, passwords and etc…
- SMS spammers and keyloggers attempt to obtain the information of their interest by sending spam SMSs that contain viruses or malware that will enable them to hack into mobile phones of their choosing.
- Misleading information, fraud, irrelevant advertisements are also some contents that can be considered as spam SMS.

In addition to the threats imposed by SMS spammers, spam SMS is simply something that is a waste of time for mobile phone users. Spam SMS results in an unclear and irrelevant SMS inbox and is ultimately a nuisance for mobile phone users.

With the product that we intend on producing, we hope to get in the way of all the unwanted situations caused by spam SMS and create a much more convenient text messaging environment for our users.

As we've stated previously, the spam SMS detection procedure will be carried out with the aid of machine learning. During the course of our research, we looked through many machine learning classifiers that were of potential use for text classification. As a result of our research, we came to the conclusion that we will be using the Naïve Bayes classifier. The Naive Bayes classifier in this project will be implemented in the Python programming language. Text preprocessing, tokenization and feature extraction (with the Bag of Words Model) will also be performed. All of the above tasks will be performed inside the server. Thus, the spam detection service will be provided to the clients(mobile phones) by the server(computer). For the adaption of our development to the mobile environment, we plan on using the React Native tool along with JavaScript. There will be three actors in our project: admin, user and registered user. The admin will have the abilities to list all users, ban/reactivate users of his/her choice from the application and list all of the users that he/she has blocked. For our product, we are giving a great deal of importance to user privacy. Thus, the admin too, will not have access to the SMS inbox of any user. People who are not registered to the application will be know as "users". Users will be able register to the application via the sign-up function. The final actor in the project is the registered user. Registered users will be able to see all SMSs in their inbox, see spam SMSs in a secondary "spam inbox", delete any SMS from any inbox, block/unblock senders that they do not want to receive text messages from, see a list of senders that they have blocked, search for SMSs from a specific sender and search for any keyword that might have occurred in any SMS that has been sent by any sender (whether the sender has been blocked or not).

3.1.3. Glossary

Stakeholders	People contributing to the project.
Python	The programming language for the implementation of the machine learning classifier.
React Native	A tool for mobile development.
MySQL Database	A database that will contain registered users and textual data.
Naïve Bayes	The machine learning classifier of our choice for the text classification process.
Bag of Words	The feature extraction model that will be used in the project.
Admin	The person who supervises the application.
User	Any user that hasn't been registered to the application.
Registered User	Any user that has been registered to the application.

3.1.4. Overview of the Document

In this section, we tried to briefly discuss the aim of our project and give an insight on how we plan on approaching the task at hand. We put forward some important definitions that we will mention frequently throughout this document, in the glossary. In the General Overview section, we will talk about our development methodology and the user characteristics. The Software Requirements Specification section will be about external interface requirements(user interfaces, hardware interfaces, etc…), functional requirements, use case diagrams of these functional requirements, non-functional requirements, performance requirements, software system attributes and finally safety requirements. The document comes to an end with our references.

3.2.Genaral Overview

3.2.1. Product Perspective

The name of our project is Spam SMS Detection Mobile Application. The end product will filter and detect spam SMSs and place them into a spam inbox. The user will be able to block/unblock senders of his/her choice. Thus, he/she will not receive SMSs from the senders he/she has blocked. Our product will provide a much cleaner and relevant SMS inbox in addition to a safe and efficient text messaging experience.

3.2.1.1. Development Methodology

We will follow a development methodology in which we will start by performing text preprocessing on the raw textual data. We plan on proceeding with tokenization and feature extraction (via the Bag of Words Model). After the text preprocessing, tokenization and feature extraction phases are complete, we will be ready to train our data set with the Naïve Bayes machine learning classifier. The React Native tool will enable us to carry our system to the mobile platform.

3.2.2. User Characteristics

3.2.2.1. User

The owner of a mobile phone, who is yet to log in to the application (regardless of whether they are registered to the application or not) is a user.

3.2.2.2. Registered User

The owner of a mobile phone, who is registered and logged in to the application is a registered user. In order to benefit from the services of the application, one must be registered and logged in.

3.2.2.3. Admin

The admin is the character with the highest privileges in the application. He/she also owns a mobile phone and is also a registered user. He/she will supervise the behavior of all registered users and basically the application as a whole. Additionally, the admin will have the privilege of being able to ban registered users who oppose a threat to either other registered users or the application itself.

3.3.Software Requirements Specification

3.3.1 External Interface Requirements

3.3.1.1. User Interfaces

User interfaces will be provided for the iOS and Android mobile operating systems.

3.3.1.2. Hardware Interfaces

For the use of our product, a mobile phone that is capable of running the up to date versions of the iOS and Android mobile operating systems is necessary.

3.3.1.3. Software Interfaces

- iOS or Android operating system
- SMPP Protocol (Short Message Peer-to-Peer)
- IP protocol (for the communication between the mobile application and the database)

3.3.1.4. Communication Interfaces

Registered users are required to be receiving services from any mobile network operator in order to be able to use our product.

3.3.2 Functional Requirements

3.3.2.1. Profile Management Use Case

Use Case:

- **Login**
- **Register**

Diagram:

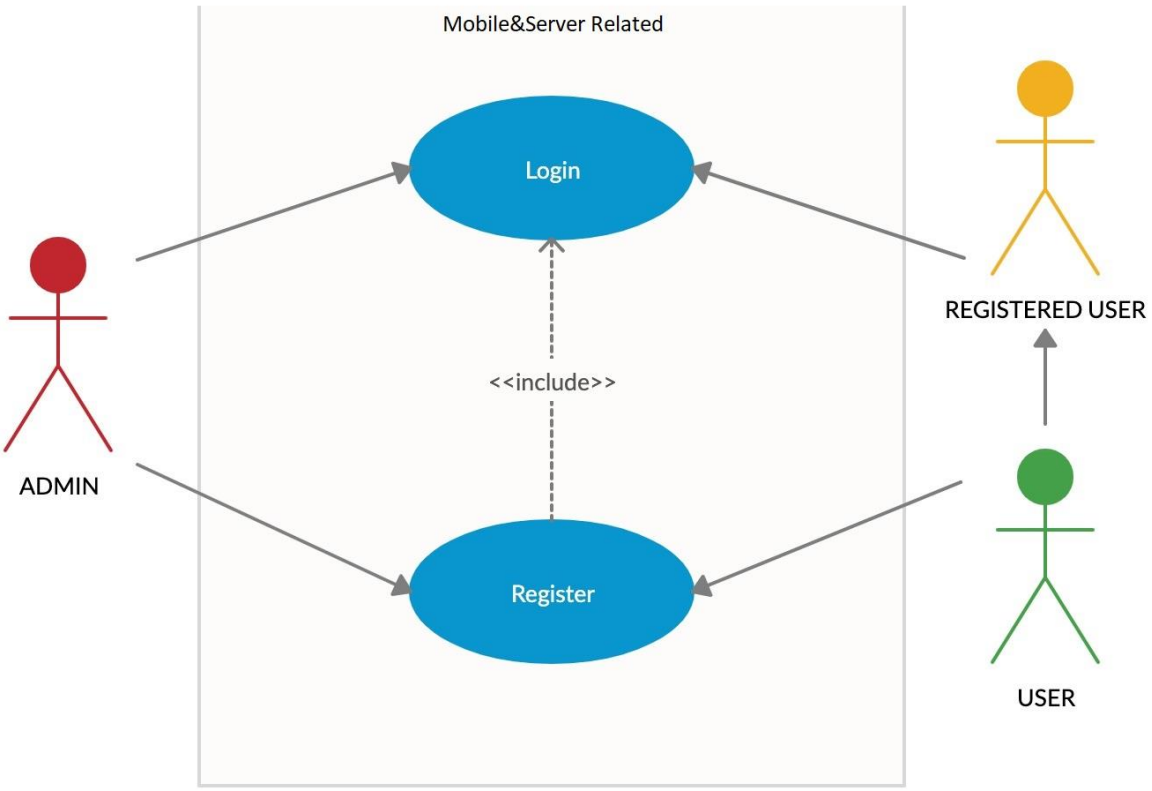

Figure 1: Profile Management Use Case

Brief Description:

Figure 1 shows the main functions performed by the admin, user and the registered user. The admin and users, all need to be registered to the application. Once registered the admin and the registered users will need to log in to the application. The “include” arrow states that one must be registered to the application in order to log in.

Initial Step by Step Description:

1. Initially, the admin must be registered to the application.
2. From then on, “users” who get registered to the application will become “registered users”.
3. The admin will need to log in to the application in order to be able to supervise the registered users, use his/her privileges and benefit from the services of the application.
4. The registered users will need to log in to the application in order to benefit from the services of the application.

3.3.2.2. Main Analysis Use Case

Use Case:

- See All SMSs
- See Spam SMSs
- Delete SMSs
- Block Spam Sender
- Unblock Spam Sender
- See Blocked Sender
- Search Contact
- Search Keyword

Diagram:

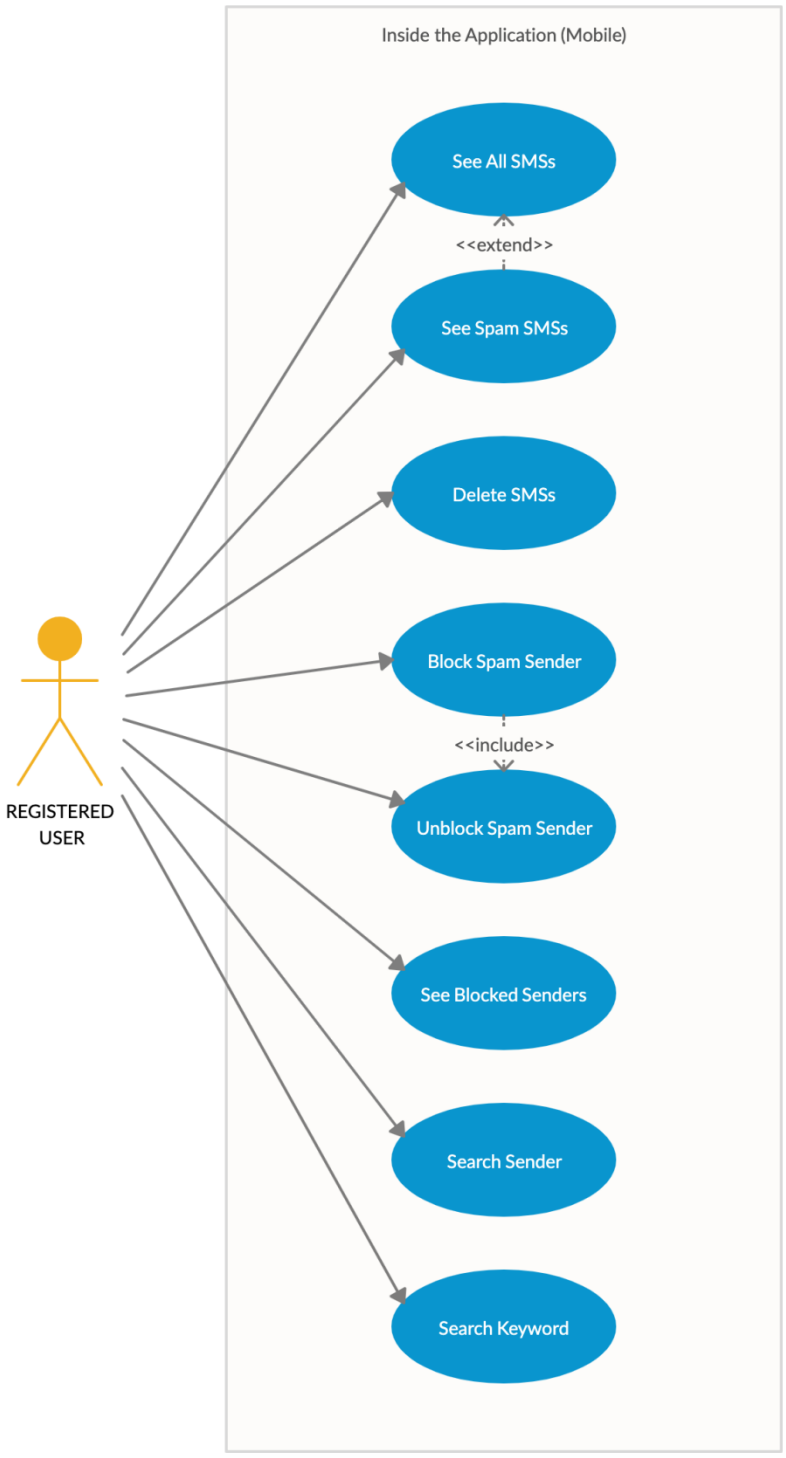

Figure 2: Main Analysis Use Case

Brief Description:

In Figure 2, the basic operations that can be performed by a registered user can be seen. A registered user has the ability to see all SMSs, see spam SMSs, delete SMSs, block spam senders, unblock spam senders, see blocked senders, search contacts and search keywords.

Initial Step by Step Description:

1. The registered user will be able to see all SMSs via a main inbox
2. Should he/she choose to see the spam SMSs he/she has received, this will also be possible via a secondary inbox in which spam SMSs will be placed by the application
3. The registered user will be able to delete SMSs from both the main inbox and the secondary inbox which contains the spam SMSs.
4. The registered user will have the ability to block a sender from whom he/she does not want to receive text messages from.
5. Step 4 is reversible, meaning that the registered user can unblock a sender that he/she had previously blocked, via the "Unblock Spam Sender" use case.
6. The registered user will be provided the ability to see a list of the senders that he/she has chosen to block.
7. If he/she would like to see the SMSs that have been sent by a certain sender, the registered user will be able to search for that sender via the "Search Sender" use case.
8. Finally, the application includes a "Search Keyword" use case. With this use case, the registered user will be able to search for a certain keyword of his/her choosing among all the SMSs received from all senders.
9. The admin is capable of performing any operation that the registered user can perform.

3.3.2.3. User Management Use Case

Use Case:

- Ban Registered User
- Reactivate Registered User
- List All Registered Users
- List All Banned Users

Diagram:

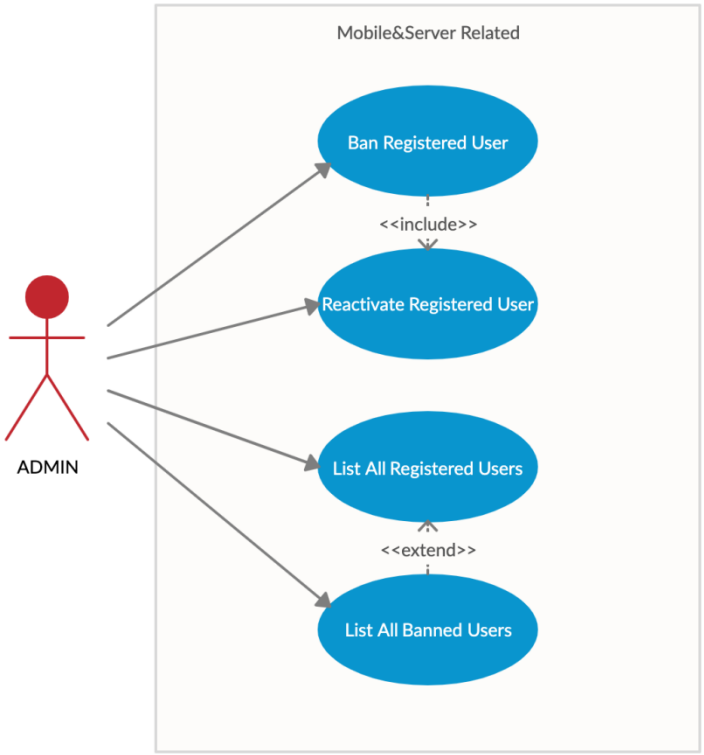

Figure 3: User Management Use Case

Brief Description:

Figure 3 shows the basic operations that can be performed by the admin. The admin is able to list all registered users, ban registered users that oppose a threat to the application or other registered users, list all banned users and reactive a registered user that he/she had previously banned.

Initial Step by Step Description:

1. The admin will have the ability to see a list of all registered users with the "List All Registered Users" use case.

2. If the admin suspects that a certain registered user is opposing a threat to other registered users or the application itself, he/she has the ability to ban that registered user from the application.
3. The admin also has the ability to see a list of the users that he/she had previously banned via the “List All Banned Users” use case.
4. Finally, the admin is capable of reactivating registered users that had been previously banned.

3.3.2.4. Personal Information Use Case

Use Case:

- Update Name
- Update Surname
- Update Phone Number
- Update Email
- Update Password

Diagram:

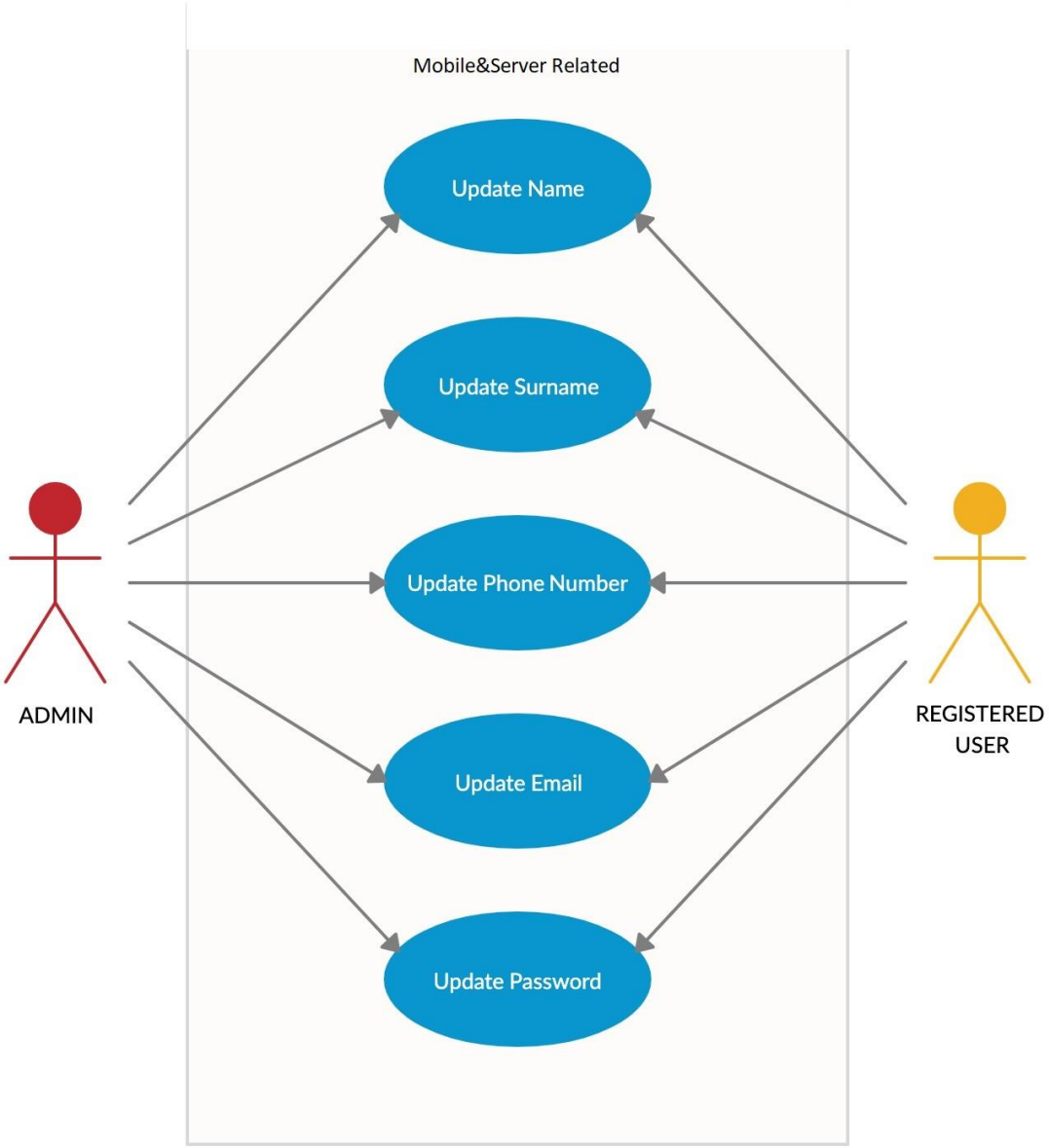

Figure 4: Personal Information Use Case

Brief Description:

Figure 4 shows the personal information update operations that the admin and registered users can perform. Name, surname, phone number, email and password are the personal information that the admin and registered users are able to update about themselves.

Initial Step by Step Description:

1. When there is a need for update regarding the personal information of the admin (these are name, surname, phone number, email and password), he/she can update this information via the Update Name, Update Surname, Update Phone Number, Update Email and Update Password use cases.
2. Similar to the admin, registered users are also able to update their personal information via the use cases mentioned in step 1.

3.3.3. Non-functional Requirements

- A white list that will keep track of words such that, if that word(s) are encountered inside the SMS, the SMS is certainly not spam.
- Since SMS departure/arrivals occur rapidly, the detection and filtration tasks should be performed rapidly as well.

3.3.4. Performance Requirements

For optimal performance, mobile phones that are capable of running the up to date versions of the iOS and Android operating systems are recommended. In addition, access to the internet is required.

Also in this section, we would like to mention our success criteria for our spam SMS detection mobile application. We aim to reach a spam detection accuracy of 85%, given that this is an academic project.

3.3.5. Software System Attributes

3.3.5.1. Portability

This product will be designed for mobile phones. Thus, the only portability that is of our interest is the portability of our product between mobile phones. Our spam SMS detection mobile application will be portable between all mobile phones that are running on either iOS or Android.

3.3.5.2. Usability

- Once inside the application, a user-friendly home page will be shown to the users, where they will be given a choice as to whether they would like to sign up or log in.
- Upon login, the registered user will be shown a menu in which he/she can perform the operations that the application has to offer. (see section 3.2.2 of this report)
- A user interface will be provided to the registered user, where he/she will be able to navigate between the services provided by the application in a clear and

understandable way (menus, buttons, colors, etc… will be utilized with the utmost care.

- Once logged in, the registered user will not have to log in every time he/she runs the application (even if he/she has killed the application in the background). This will go on until the registered user chooses to log out of the application.

3.3.5.3. Adaptability

This product will not necessitate any adaptability requirements.

3.3.5.4. Availability

We will be aiming to see our application on the App Store (for iOS) and the Google Play Store (for Android). Again, access to the internet is an obligation.

3.3.5.5. Scalability

We are aware of the increasing work load that may arise for our application, given that the demand for it increases. In such a case, we have some solutions that we can propose:

- Utilizing a bigger server (scaling up) or increasing the number of servers (scaling out)
- Increasing CPU power
- Adding more memory, thus storage to the present one

On the long run, scaling up has a risk of creating bottlenecks between CPU and memory and may not be as preferable as scaling out when the cost-to-benefit ratio is taken into consideration.

3.3.6. Safety Requirements

As a project team, the safety and privacy of our registered users is of utmost significance to us. Therefore, the SMSs of all registered users, along with the content of these SMSs, will not be visible to any other registered user or even the admin himself/herself. For the most safe and private experience with our application, we will be strictly advising registered users on keeping their passwords to themselves and confidential.

4. Software Design Description

4.1.Introduction

This report intends to inform and guide the user regarding the design of our software. The software design description and working procedures of our spam SMS detection mobile application will be presented throughout the report, in detail.

4.1.1. Purpose

The purpose of our Spam SMS Detection Mobile Application software is to maintain a safe, private and relevant text messaging environment in the world of telecommunication. None of us want to receive SMS messages that are not relevant with respect to our interests. The proposed software will detect and filter spam text messages received by the user, and allow for a more relevant and understandable SMS inbox. In addition, the personal and confidential information of the users of our software will be protected against the malicious intentions of SMS spammers, who are constantly trying to unwillingly access and obtain this information. With that said, we hope to contribute to the text messaging experience of our users in a way that is most user-friendly and convenient for them.

4.1.2. Scope

The spam SMS detection model will be based on the Naive Bayes classifier algorithm, which will be implemented in the Python programming language. Text preprocessing, tokenization and feature extraction techniques will be performed in cooperation with the Naive Bayes algorithm. The mobile application will be developed inside a computer environment, with the help of the React Native tool and JavaScript. The mobile application will rely on a server (computer) to carry out rapid spam SMS detection and filtration tasks.

Registered users will be given access to two different SMS inboxes: one that contains all SMSs and one that contains solely the spam SMSs. The contents of a spam SMS will be hidden, unless the registered user would like to see it. For each spam SMS, registered users will be given a choice:

- Delete the spam SMS.
- Delete the spam SMS and block the sender.

The user interface will include a settings page where registered users will be able to get help regarding the use of the application, update their personal information and see a list of the senders they have chosen to block. Here, they will be able to unblock senders that they

had previously blocked. The user interface will be designed in a clear and user-friendly fashion that will allow registered users to benefit from the services of our application with ease.

There will be a database within the server. The database will contain:

- The registered users and their personal information that they have given to register to the system.
- The SMSs of all registered users.

All data inside the database will be kept strictly confidential for the privacy and security of our registered users.

4.1.3. Glossary

TERM	DEFINITION
User	Any character that is yet to register to the system.
Registered User	Any character that is registered to the system and thus, is able to benefit from the services of the system.
Admin	The character that supervises the system and is also a registered user himself/herself.
Spam SMS	Any SMS that is not of the registered user's interest or opposes a threat to the registered users personal/confidential information.
Database	The database where the registered users of the system are stored (with their personal information and SMS inboxes)
Block Diagram	A diagram that displays the components of the software in blocks.

UML Diagram	A modeling language that is used in the software development process.

4.1.4. Overview of the Document

Part 1 of this document has served as an introduction to the software design description of our project. Part 2 will focus on the problem that the project intends on resolving and the architecture of the system. Part 3 will concentrate on the block diagrams, which have been drawn based on the information gain that we have acquired throughout the Software Requirements Specification period of our project.

4.1.5. Motivation

As also mentioned in the Literature Review and Software Requirements Specification documents, our motivation for this project is to provide a safe and convenient text messaging environment for mobile phone users. The popularity and widespread use of SMS messaging has unfortunately created an opportunity for SMS spammers to exploit from mobile phone users and intrude their privacy. SMS spammers are on the quest to obtain the personal/confidential information (such as credit card numbers, passwords, contact lists, ...) of mobile phone users via the malware that they try to place inside their mobile phones. Links inside spam SMSs oppose a threat to mobile phone users because they may direct mobile phones to viruses or malware. In addition to SMS spam being a threat, it is also a waste of time and space for mobile phone users. It is possible to say that we all want to receive text messages that are legitimate with respect to our interests or potential interests. The mobile application we intend on developing is bound to create a safe, private and relevant SMS inbox and offer a convenient text messaging environment to all our registered users

4.2. Design Overview

4.2.1. Description of the Problem

The problem that served as a driving factor for this project was the increase in the SMS traffic throughout the world. This increase brought along an increase in SMS spam. We carried out a thorough research during the Literature Review for our project and gained valuable information regarding what SMS spam is and how it can be handled. The negative effects of SMS spam have been discussed in detail, above in Part 1 of this document. With the mobile application that we are developing, we aim to get in the way of SMS spam and protect

our registered users from the threats opposed to them by SMS spammers and keyloggers while allowing for a convenient and legitimate text messaging environment for everyone.

4.2.2. Technologies Used

The Naïve Bayes classifier for the spam SMS detection task will be implemented in the Python programming language. We will use Jupyter Notebook as the program development environment for Python. Once the spam SMS detection and filtration model is ready, we will use the React Native tool along with JavaScript to develop our spam SMS detection mobile application.

4.2.3. Architecture Design

Before proceeding any further, we would like to restate some of the contents that are inside our Literature Review and Software Requirements Specification documents in order to briefly discuss our dataset, and spam detection/filtration model. The proposed spam SMS detection mobile application will be developed for the English language, as this will allow us to appeal to the widest range of users. Our dataset will be the SMS Spam Collection Dataset from the UCI Machine Learning repository. [16] For the generation of our model:

- Text preprocessing will be performed on the raw data,
- Tokenization of words will be done,k
- Feature extraction and vectorization (via the Bag of Words method in Python) will be done,
- The dataset will be proportioned into 70:30 (in terms of training/testing)
- 70 percent of the data will be used to train the model with the Naïve Bayes machine learning classifier,
- 30 percent of the data will be used to test the trained model.

The spam SMS detection model will be located inside a computer environment **(which will be our server)** and spam detection/filtration will be performed here. This will ultimately necessitate an efficient utilization of a database, which will also be located inside the server. There will be a client-server model in the system, where all mobile phones (that are using our application) will be clients that are acquiring services from the server. The SMSs of all registered users will stored inside the database, as they are received on mobile phones.

Each SMS will be accessed from the database and put through our spam SMS detection model at the server. The value of the IsSpam Boolean variable (will be provided in the Class Diagram) for each SMS will be updated depending on the output of the spam SMS

detection model. Based on the value of the IsSpam variable of a given SMS, that SMS will be shown to the registered user (who will be using the application on his/her mobile phone) as either spam or legitimate. Spam SMSs will be placed inside the Spam Box while legitimate SMSs will be placed inside the All SMSs box.

The general structure of our spam SMS detection model has been given above. Just for clarification, our application will not be designed for messaging (conversation) purposes. The application will be designed for detecting potential spam content inside the text messages that our registered users will have received. Thus, registered users will not be able to generate any kind of reply to the SMSs that they have received, by using our application.

4.2.3.1. Class Diagram

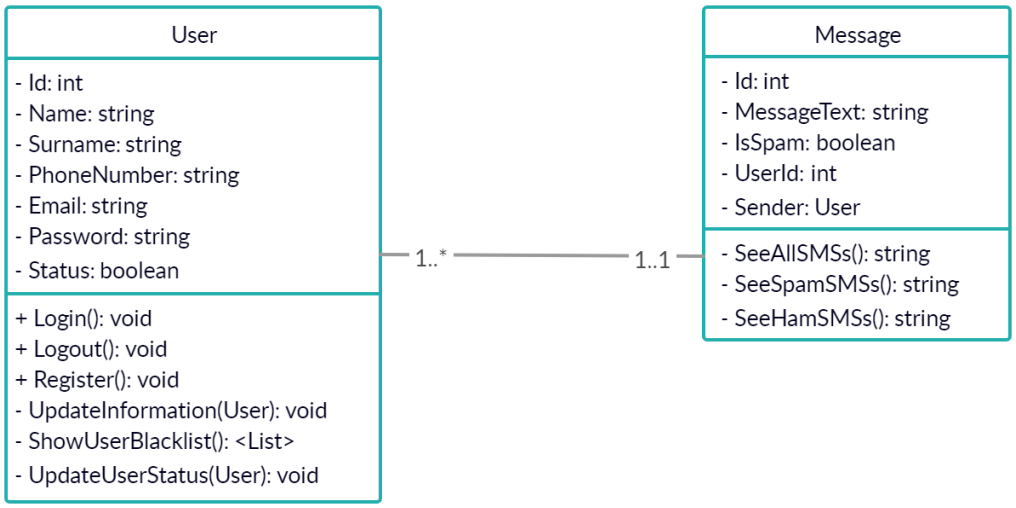

Figure 1: The class diagram displaying the functions of the spam SMS detection mobile application.

Figure 1 shows the User and Message classes that are necessary for the operations of our spam SMS detection mobile application. First, let's start with the User class. The Register() function will enable the user to register to the system using his/her username, phone number and email address. This information will be stored in the Username, PhoneNumber and Email variables respectively, for each user. During registration, the user will pick a password for himself/herself which will be stored in the Password variable.

Once registered to the system, the registered user will be able to login to the system with the Login() function, using his/her username and password. With the UpdateInformation(User) function, the user will be able to update his/her personal information. The ShowUserBlacklist() function is for the user to view the list of the registered users that he/she has blocked. The status variable is to distinguish whether a registered user is

banned from the system (by the admin) or not. If the Status variable is 1 for a given registered user, that registered user is active. If the value of the Status Variable is 0 for a given registered user, this means that the registered user has been banned from the system by the admin. Thus, the UpdateUserStatus(User) function is for the admin to update the Status variable of the registered users. Finally, the Logout() function will enable the registered user to logout from the system. (For clarification, the ID variable will not be asked from the user during registration and will only be utilized inside our database as the PRIMARY KEY of the table that stores the registered users.)

The second class is the Message class. Similar to the ID variable in the User class, each message will also be stored in our database with its own ID. This ID variable will serve as the PRIMARY KEY of the table that will store all SMSs in the system. The MessageText variable is the variable that will hold the contents of a given SMS. The UserId variable will hold the ID of the user who received that particular SMS. The Sender is a User object and will contain the information of the sender of that particular SMS. If the sender is not a registered user, he/she will be automatically assigned a guest username. The IsSpam variable will be used to distinguish whether a given SMS is spam or not. If the value of the IsSpam variable is 0, this means that the SMS is legitimate. If the value of the IsSpam variable is 1, then that SMS is spam. The SeeAllSMSs() function will enable the registered user to see all of the SMSs in his/her inboxwhile the SeeSpamSMSs() function will only view the spam inbox. Lastly, the SeeHamSMSs() function will show the registered user, only the legitimate SMSs.

4.2.3.2. Home Page

Actor(s): Admin, Registered User

Precondition: The application must be launched and running.

Upon launching the application, the registered user will be face to face with the home page. The home page is actually the page where his/her inbox is presented to the registered user. There will be buttons that can direct the registered user to the spam inbox page and settings page. The registered user will be able to benefit from the services of the system from the home page (main inbox page), spam inbox page and settings page.

Basic Sequence:

1. After launching the application, the home page (inbox page) will be visible to the registered user.

2. The sender (will be shown with his/her name in the contact list or just his/her phone number, given that he/she is not in the contact list) of each SMS will be seen as a label on the SMS. The first few words of the contents of the text messages will be visible as a preview.
3. Clicking on an SMS will open the contents of it.
4. In the home page, each SMS will have a trash can icon next to it. The registered user can choose to delete a given SMS by clicking on this trash can.
5. Via the buttons on the home page, the registered user can navigate between the home page (inbox page), spam inbox page and the settings page.

4.2.3.3. Activity Diagram

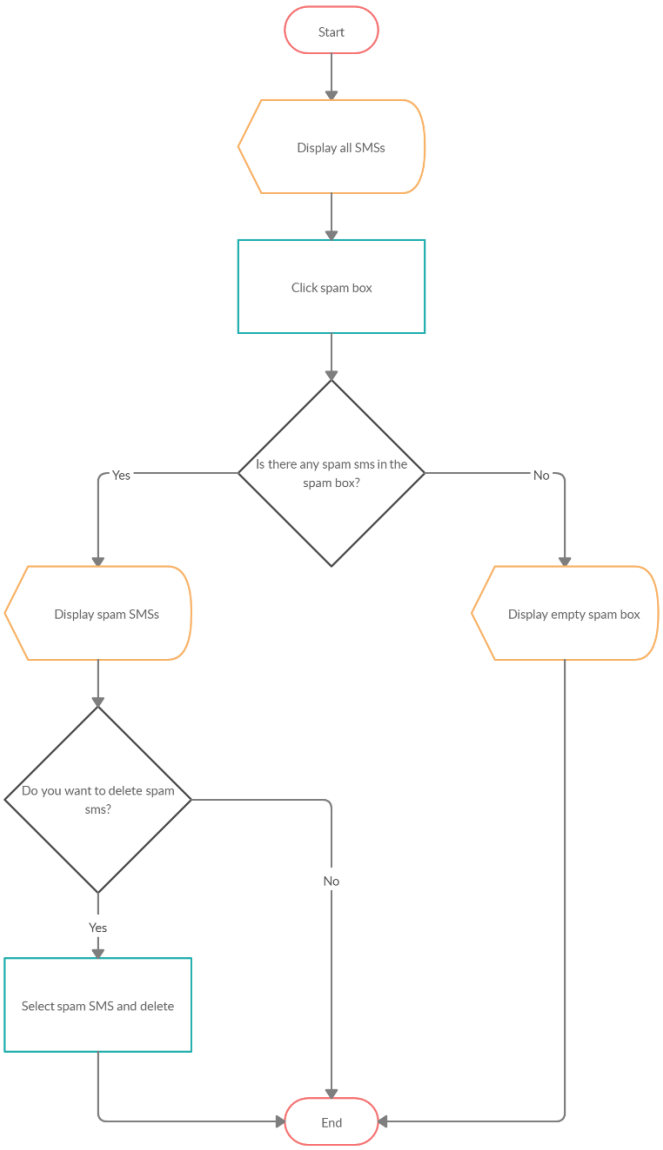

Figure2: The activity diagram for the spam SMS detection mobile application.

Upon launching the application, the registered user will be face to face with the home page where he/she can see all of the SMSs that he/she has received. Once he/she clicks the spam inbox button, he/she will be directed to the spam SMS inbox. If there are spam SMSs present at the moment, they will be displayed. Otherwise, an empty spam inbox will be displayed and the registered user will exit the system. In the case where spam SMSs are present, a choice as to whether he/she wants to delete a given spam SMS will be given to the registered user for each spam SMS. This will be possible through the trash can icon that will

be placed next to each spam SMS. Once the registered user has deleted the SMS(s) that he/she had chosen to delete, he/she will exit the system.

4.2.3.4 Project Plan

The Gantt chart of our work plan is given below:

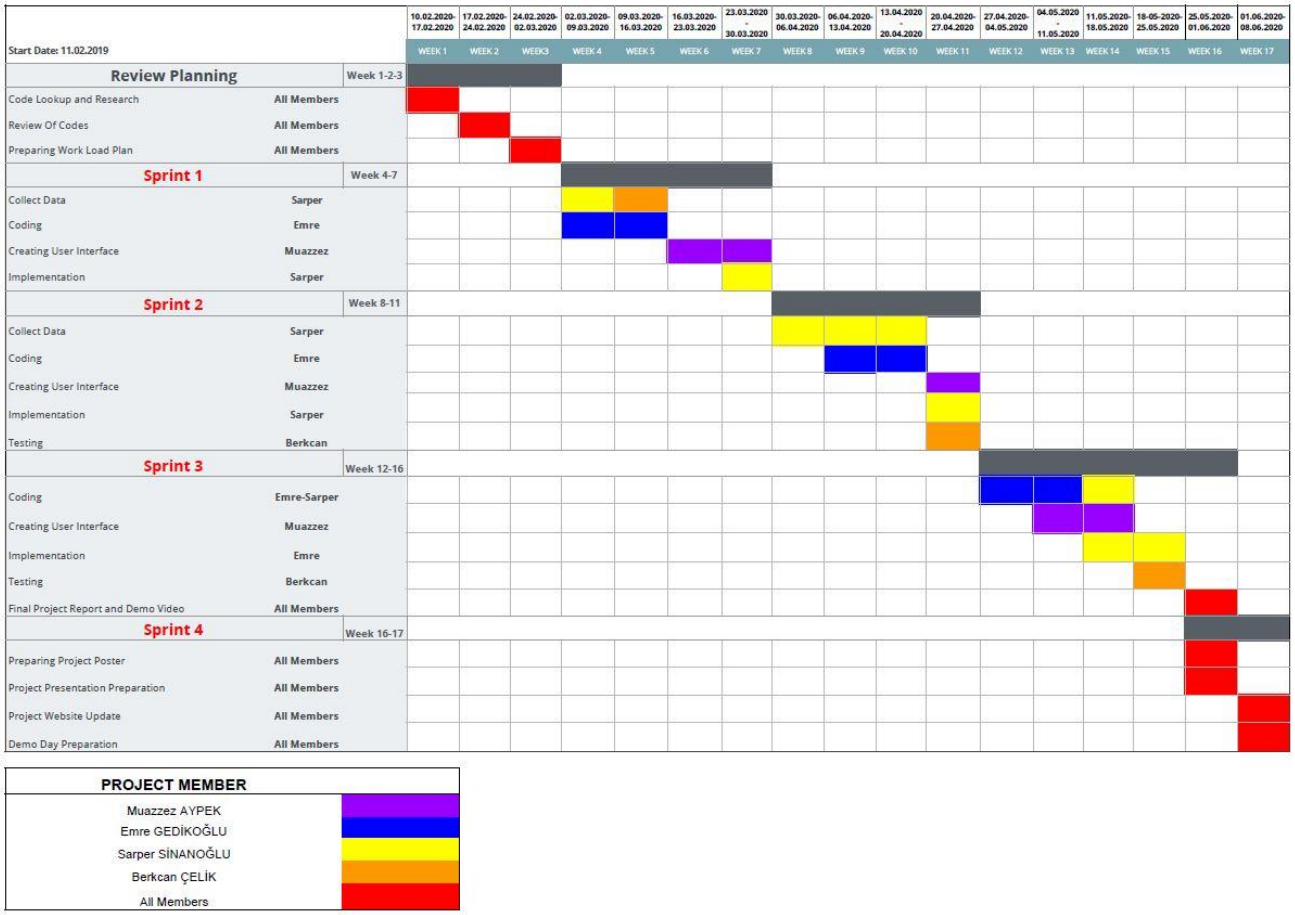

Figure 3: The Gantt chart of the project work plan.

There will be various tasks and assignments throughout the software development process. As team members, we will participate in all of these tasks and assignments together, as a team. However, dividing the workload between team members is bound to increase our efficiency and raise self-awareness for us all. We have shown the dividing of the workload on the above Gantt chart and aim to follow the given project work plan.

Code Lookup and Research: The task in which we will research new algorithms, code snippets and techniques that are of potential use for our project.

Review of Codes: The task in which we will review and reconsider our code for accuracy and compatibility.

Preparing Workload Plan: The preparation of the weekly and monthly workload distribution between team members. This will be updated as the tasks and assignments are completed.

SRS and SDD Finalization: Making updates to the SRS and SDD documents based on change of needs and design renewals.

Preparing Test Plan: The preparation of the test cases in the Test Design Specifications document once they are properly designed and classified.

Testing and Maintenance: Checking for errors and testing undesired conditions.

Software Design: User interface and functionality design based on user requirements.

Software Implementation and Testing: Forming the components of the software and testing their functionality.

Testing and Release: Final tests for functionality, fixing bugs and making possible improvements. Release of the final product.

Final Project Report and Demo Video: Recording a video of the developed software in demo format. The video will then be uploaded to Youtube.

Project Presentation Preparation: Preparation of the presentation slides and presentation rehearsals.

4.2.4. System Operation

For each spam SMS, the registered user will be given a choice as to whether he/she wants to simply delete the spam SMS or block the sender also. This functionality will be provided with the aid of a pop-up window. The window will include two buttons with one labeled “Delete” and the other labeled “Delete and Block”. In the future, the registered user may want to unblock a sender whom he/she had blocked previously. This functionality is also provided. From the home page, navigation to the settings page is possible through an icon that represents settings. At the settings page, the registered user will be offered a button. By pressing this button, the registered user will be able to see a list of the senders that he/she had blocked in the past. Next to each sender, an icon will be present. Clicking on that icon will unblock that sender.

In addition to registered users being able to block each other, the admin has the privilege of banning registered users of his/her choice, from the system. If the admin suspects

any threats from a given registered user, he/she will update the Status variable of that registered user via the UpdateUserStatus(User) function. Once the value of the Status variable for a given registered user has been updated to 0 by the admin, that registered user will be banned from the system and will not be able to benefit from the services provided by the system.

Finally, changes in the personal information of registered users need to be known and updated in the database. Username, password, phone number and email changes will be handled with a personal information button (that will direct the registered user to a page where he/she can update the changes in his/her personal information) in the settings page. The registered user will be able to update his/her personal information there. Updates in the personal information of the registered user will be automatically transferred to our database on our server.

4.3.Use Case Realizations

4.3.1. Spam SMS Detection Mobile Application System

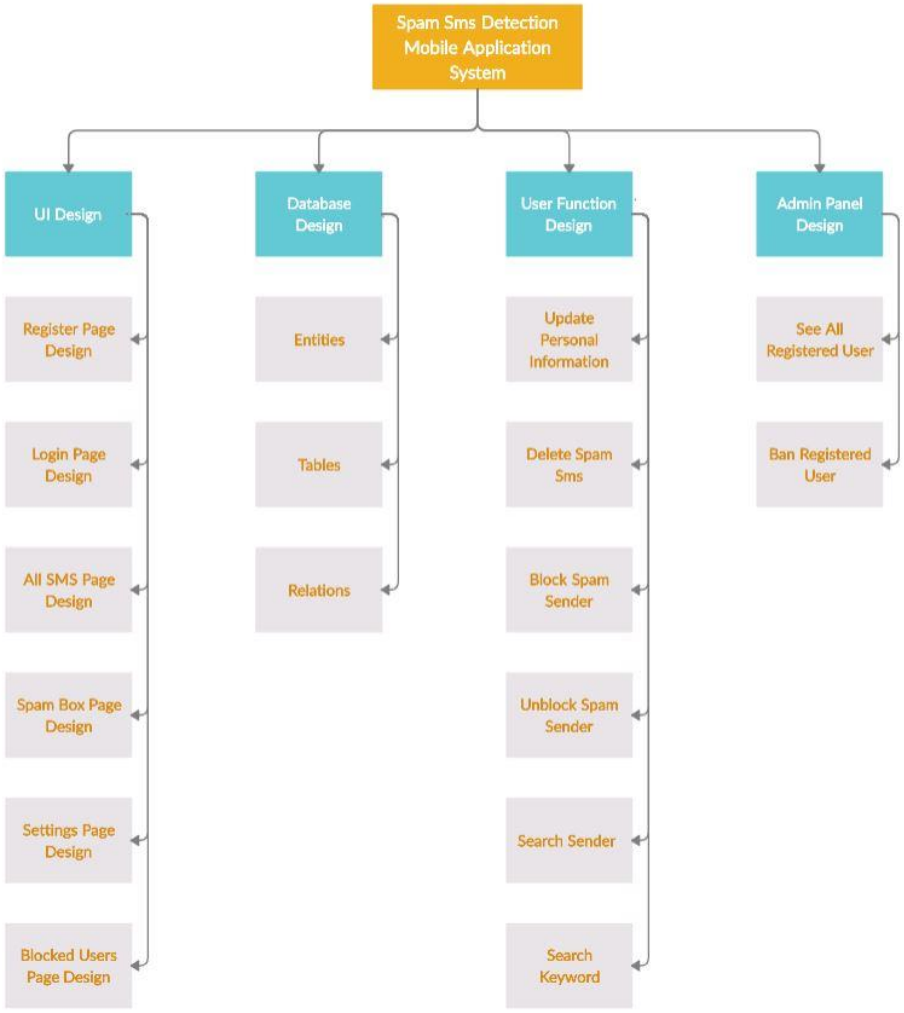

Figure 4: The Spam SMS Detection Mobile Application System

Above, Figure 4 displays the block diagram of our spam SMS detection mobile application system. The block diagram is composed of four main components (UI Design, Database Design, User Function Design and Admin Panel Design) and their sub-systems. Each of the main components is discussed below:

4.3.1.1. UI Design

The User Interface (UI) design is made to explain the interactions between the user (not registered) / registered user and the system. The UI design has six sub-systems:

- **Register Page Design:** Designed for users to register to the system using their personal information. Users who have registered to the system will become registered users.

- **Login Page Design:** Designed for registered users to login to the system with their usernames and passwords.
- **All SMS Page Design:** Designed for registered users to see all of the legitimate text messages that they have received.
- **Spam Box Page Design:** Designed for registered users to see all of the spam text messages that they have received.
- **Settings Page Design:** Designed for registered users to update their personal information, see the list of other registered users that they have blocked, get help regarding the use of the application and possibly delete their account (if they do not wish to use the application anymore).
- **Blocked Users Page Design:** Designed for registered users to see the list of other registered users whom they have blocked.

4.3.1.2. Database Design

The Database Design is made in order to efficiently plan the storage of the data that is necessary for the operation of the system. The three sub-systems for the database design are: entities, tables and relations. We will have two entities: User and Message. The tables used for storing the User and Message entities will be related to each other.

4.3.1.3. User Function Design

The User Function Design describes the possible functions of the registered user. The six sub-systems indicate that registered users are able to update their personal information, delete a selected spam SMS, block a selected spam SMS sender, unblock a spam SMS sender that they had previously blocked, search for a sender of their choice and search for a certain keyword.

4.3.1.4. Admin Panel Design

The Admin Panel Design is used to explain the functions of the admin. The admin is capable of performing all of the functions that are done by the registered user (as also stated in the Software Requirements Specification document). In addition to these functions, the admin has two privileges. These privileges are shown with the sub-systems of the Admin Panel Design: See All Registered Users and Ban Registered User. The admin is able to view a list of all registered users (with their personal information such as phone number and email) along with their Status variable. Furthermore, the admin is able to ban registered users from the system by updating their Status variable to 0. Inversely, the admin can reactivate the account of a previously banned registered user by updating his/her Status variable to 1 again.

4.3.2. User Interface

This section of our report will concentrate on the buttons and layouts of the user interface pages. The design of these pages has been made in order to maintain a user-friendly user interface. The pages are the: Register Page, Login Page, All Messages Page, Spam Box Page, Settings Page and the Blocked Users Page.

4.3.2.1. Register Page

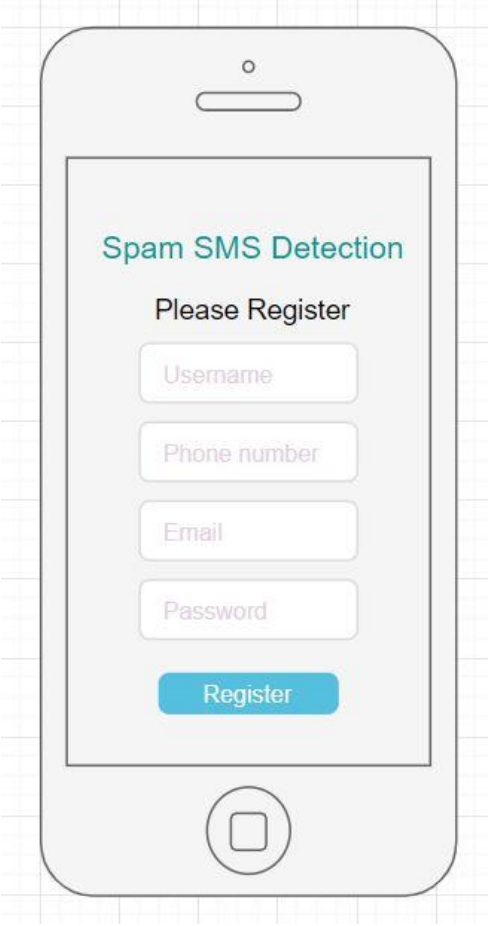

Figure 5: The Register Page.

The Register Page is presented above, in Figure 5. This page has been designed for non-registered users to register to the system, and thus become registered users. During registration, the users select a username and password for themselves. Their username and password pair will be used by registered users to login to the system and benefit from its services. Phone number and e-mail are the two other pieces of information that are required for registration. Registered users will be able to change/update their username, password, phone number and/or email in the future; if necessary.

4.3.2.2. Login Page

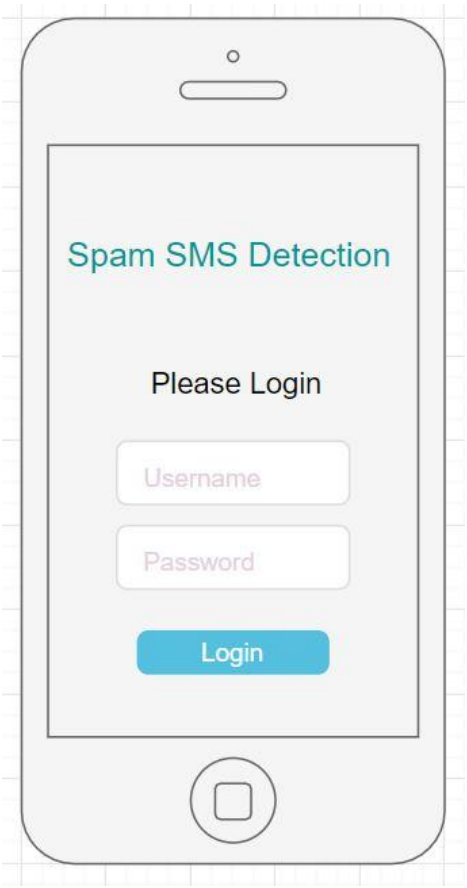

Figure 6: The Login Page.

Figure 6 displays the Login Page. This page is for registered users to login to the system and start benefiting from its services. Registered users who are viewing this page will have already registered to the system and decided on their username and password. They will enter their username and password, and click on the “Login” button. Then, they will be logged in to the system.

4.3.2.3. All Messages Page

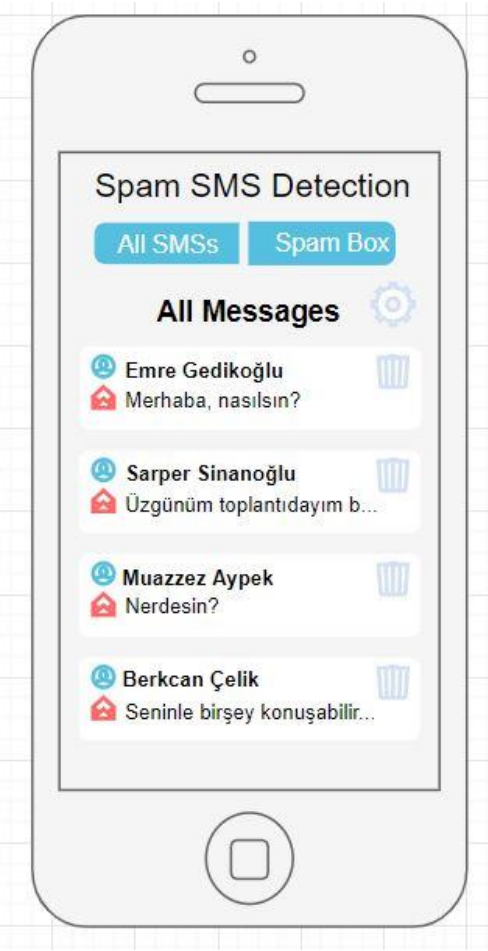

Figure 7: The All Messages Page.

Above, Figure 7 shows the All Messages Page. Upon logging in, this is the page that the registered users will be face to face with. Thus, the All Messages Page will also serve as the home page of the application. In this page, the registered users will be able to view the legitimate text messages that they have received. The first few words of each SMS will be shown as a preview of that SMS. Clicking on an SMS will open it and view its full contents. Next to each SMS, there is a trash can icon. Clicking on the trash can icon that is next to a particular SMS will delete that SMS. Above the "All Messages" text, there are two buttons: All SMSs and Spam Box. The "All SMSs'' button is used to navigate to the All Messages Page while the "Spam Box" button is used to navigate to the Spam Box Page (will be presented below). Next to the "All Messages" text, there is a gear icon. Clicking on this gear icon will take the registered user to the Settings Page (which will also be presented below). These are the functionalities of the All Messages Page.

4.3.2.4. Spam Box Page

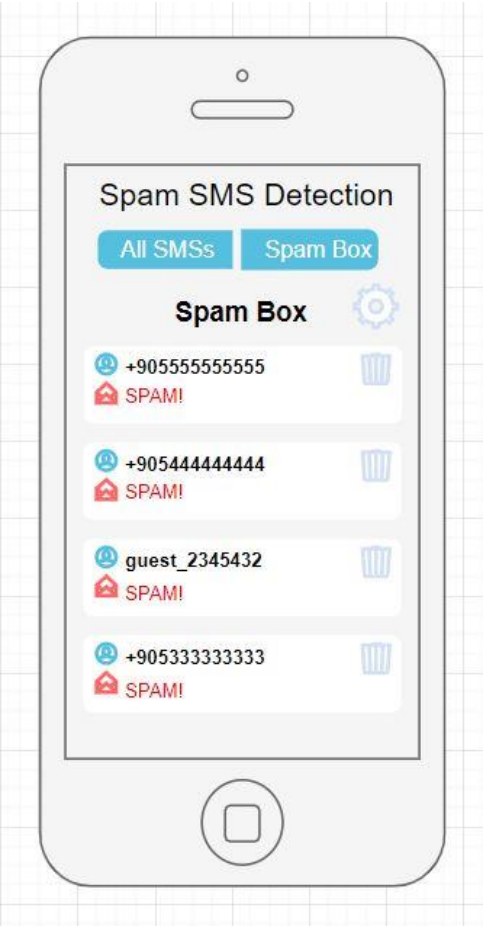

Figure 8: The Spam Box Page.

Figure 8 shows the Spam Box Page. This page has been designed for registered users to be able to view the spam text messages that they have received. Navigation between the All Messages Page and the Spam Box Page is possible with the help of the "All SMSs" and "Spam Box" buttons. As mentioned previously, the "All SMSs" button will take the registered user to the All Messages Page, while the 'Spam Box" button will take the registered user to the Spam Box Page. Unlike legitimate text messages, the first few words of spam SMSs are not displayed as a preview. Instead, the string "SPAM!" is displayed for each spam SMS. Similar to legitimate SMSs, each spam SMS has a trash can icon next to it. The functionality of the trash can icon that is next to spam SMSs is a bit different than the functionality of the one next to legitimate SMSs. Clicking on the trash can icon that is next a particular spam SMS will open up a pop-up window. This pop-up window will give the registered user a choice as to whether he/she wants to simply delete that particular spam SMS, or block its sender in addition to deleting it. Finally, the gear icon that is next to the "Spam

Box” text is used to navigate to the Settings Page, which is presented below in section 3.2.5 of this document.

4.3.2.5. Settings Page

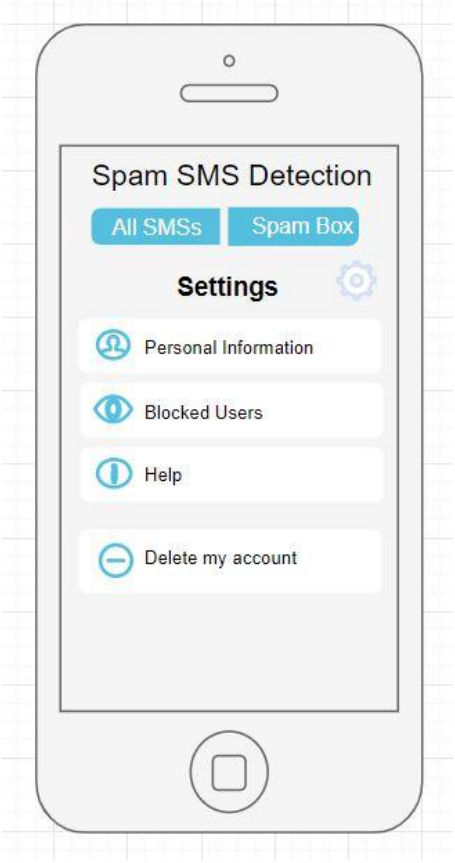

Figure 9: The Settings Page.

The Settings Page, which is given in Figure 9, has been designed to enhance the functionality of our spam SMS detection mobile application. Similar to the All Messages Page and Spam Box Page; the “All SMSs” button, “Spam Box” button and the gear button (icon) are present. As mentioned previously, these three buttons are for navigation between the All Messages Page, Spam Box Page and the Settings Page. The Settings Page contains four different options: Personal Information, Blocked Users, Help and Delete My Account. Clicking on the “Personal Information” option will take the registered user to a page where he/she can change or update his/her personal information such as username, password, phone number and/or e-mail. The “Blocked Users” option will navigate the registered user to a page where he/she can view a list of the registered users that he/she had previously blocked. The “Help” option is for registered users to obtain information regarding the use of the application. Lastly, the “Delete My Account” option will enable registered users, which do not want to benefit from the services of the system anymore, to delete their accounts.

4.3.2.6. Blocked Users Page

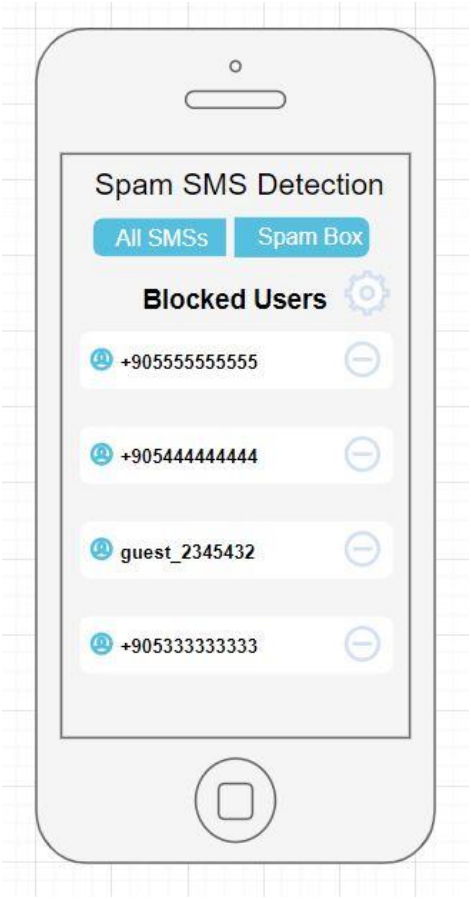

Figure 10: The Blocked Users Page

Above in Figure 10, the Blocked Users Page is given. The Blocked Users Page is reachable through the Settings Page. To navigate back to the Settings Page, the registered user needs to click on the gear icon that is next to the “Blocked Users” text. At the Blocked Users Page, the list of the registered users that he/she had previously blocked is shown to the registered user. Next to each blocked registered user, there is an icon. By clicking on the icon that is next to a particular blocked registered user, that registered user can be unblocked. These are the functionalities of the Blocked Users Page.

5. References

[1] Nagwani, N. K., & Sharaff, A. (2017). SMS spam filtering and thread identification using bi-level text classification and clustering techniques. *Journal of Information Science*, *43*(1), 75–87.

[2] D. Delvia Arifin, Shaufiah and M. A. Bijaksana, "Enhancing spam detection on mobile phone Short Message Service (SMS) performance using FP-growth and Naive Bayes Classifier," *2016 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*, Bandung, 2016, pp. 80-84.

[3] Bakliwal, Aditya & Agarwal, Shubhangi & Mehndiratta, Pulkit. (2018). A Comparative Study of Spam SMS Detection Using Machine Learning Classifiers. 1-7.

[4] M. Jameel, Noor. (2018). SMS spam detection using association rule mining based on sms structural features. Journal of Theoretical and Applied Information Technology. 96.

[5] Kaya, Yilmaz. (2018). Spam SMS'lerin filtrelenmesinde yeni bir yaklaşım: Motif örüntüler.

Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji. 6. 436-450.

[6] Bäckman, D. (2019). EVALUATION OF MACHINE LEARNING ALGORITHMS FOR SMS SPAM FILTERING (Dissertation).

[7] Pradeep Kumar Roy, Jyoti Prakash Singh, Snehasish Banerjee. (2019). Deep learning to filter SMS Spam. Future Generation Computer Systems. Volume 102. Pages 524-533.

[8] Akshay Divakar, Sitaraa Krishnakumar. (2019). SMS Spam Detection using Tokenization and Feature Engineering. International Journal of Recent Technology and Engineering (IJRTE).

Volume-8 Issue-3. Pages 6805-6807.

[9] Tokenize Words and Sentences with NLTK. [Online].

Available:https://www.guru99.com/tokenize-words-sentences-nltk.html.

[10] Sable, M.S., & Kalavadekar, P.N. (2016). SMS Classification Based on Naïve Bayes Classifier and Semi-supervised Learning.

[11] Ham or Spam? SMS Text Classification with Machine Learning. [Online].

Available:https://towardsdatascience.com/sms-text-classification-a51defc2361c.

[12] Popovac, Milivoje & Karanovic, Mirjana & Sladojevic, Srdjan & Arsenovic, Marko & Anderla, Andras. (2018). Convolutional Neural Network Based SMS Spam Detection. 1-4. **[13]** Kim, Hwa-Yeon & Lee, Jinsu & Yeo, Na & Astrid, Marcella & Lee, Seung-Ik & Kim, Young-Kil. (2018). CNN based Sentence Classification with Semantic Features using Word Clustering. 484-488.

[14] Implementing a Multinomial Naive Bayes Classifier from Scratch with Python. [Online]. Available: https://medium.com/@johnm.kovachi/implementing-a-multinomial-naive-bayesclassifier-from-scratch-with-python-e70de6a3b92e.

[15] Introduction to Logistic Regression. [Online].
Available: https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148.

[16] Logistic regression – introduction and advantages. [Online]. Available:https://www.oreilly.com/library/view/statistics-formachine/9781788295758/a56cd46a-b5db-44a8-962e-c0528f07bb9b.xhtml.

[17] Makine Öğrenimi Bölüm-4 (Destek Vektör Makineleri). [Online]. Available: https://medium.com/@k.ulgen90/makine-%C3%B6%C4%9Frenimib%C3%B6l%C3%BCm-4-destek-vekt%C3%B6r-makineleri-2f8010824054.

[18] Abdalla, Sevgi & Erdoğmuş, Şenol. (2014). Destek Vektör Makineleriyle Sınıflandırma Problemlerinin Çözümü için Çekirdek Fonksiyonu Seçimi. Eskişehir Osmangazi Üniversitesi İktisadi İdari Bilimler Fakültesi Dergisi. 9. 175-198.

[19] Jain, Gauri. (2017). Spam Detection on Social Media Text. INTERNATIONAL JOURNAL OF COMPUTER SCIENCES AND ENGINEERING. 5.

[20] P. Sethi, V. Bhandari and B. Kohli, "SMS spam detection and comparison of various machine learning algorithms," *2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN)*, Gurgaon, 2017, pp. 28-31.

[21] Why Random Forest is My Favorite Machine Learning Model. [Online]. Available: https://towardsdatascience.com/why-random-forest-is-my-favorite-machinelearning-model-b97651fa3706.

[22] Verikas, Antanas & Vaiciukynas, Evaldas & Gelzinis, Adas & Parker, James & Olsson, M. Charlotte. (2016). Electromyographic Patterns during Golf Swing: Activation Sequence Profiling and Prediction of Shot Effectiveness. Sensors. 16. 592.

[23] Nazirova, S. (2011). Survey on Spam Filtering Techniques. *Communications and Network, 3*, 153-160.

[24] A Gentle Introduction to the Bag-of-Words Model. [Online].
Available: https://machinelearningmastery.com/gentle-introduction-bag-words-model/.

[25] P. Sethi, V. Bhandari and B. Kohli, "SMS spam detection and comparison of various machine learning algorithms," *2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN)*, Gurgaon, 2017, pp. 28-31.

[26] SMS Spam Collection Data Set. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/sms+spam+collection.