COMPUTING
WITH
PLYMOUTH
UNIVERSITY

School of Computing and Mathematics

PRCO303
Final Stage Computing Project

BSc (Hons) Computer security

R.D.H.K Madusanka
10638111

Spam Filtering Application

2019/2020

1. Acknowledgment

This project is like a bridge between theoretical and practical world. This enabled me to extend my knowledge and gain practical skills. First, I would like to thank University of Plymouth for providing us with this great opportunity within the academic degree.

I deeply appreciate my project supervisor Mr. Chamindra Attanayaka for being the most helpful and supportive person for me throughout this project. Without his guidance I may not have been able to complete this project. His continuous support and motivation kept me on track during the tough times.

Finally, I would be grateful to my parents, friend Tharusha Kudagala and people who did not mentioned here but whom that answered to my questions on groups and forums for being supportive and for the motivation they provided me.

2. Abstract

This report describes a computer security project to develop a method to detect spam emails and block them with the help of people which use email accounts. An email user can rate received email by choosing whether that email is spam or not spam according to their personal perspective. Eventually, millions of user ratings are used to improve the accuracy of spam detection by calculating the probability of a future email being spam or not spam.

The report commences with an introduction chapter which is focused on introducing the project by explaining the current statistics on spam and basic principles and techniques that are going to be used in this project. After defining scope of the project next comes the deliverable section that describes the deliverables of the project. Then the background chapter explains history of email, spam and introduce that while describing existing technologies and techniques in the present time. Literature review, business case and motivation come next then approach and method sections describe how the project was planned and carried out until the completion. Finally, in the postmortem outcome of the project is evaluated.

Reference, bibliography and appendices are listed at the end of the with relevant diagrams and other required official documents.

3. Table of Content

4. List of figures and tables

5. Introduction

In 2018, approximately 281 billion (*Daily number of e-mails worldwide 2023 | Statista*, 2019) e-mails were sent and received every day worldwide. This figure is projected to increase to over 347 billion (*Daily number of e-mails worldwide 2023 | Statista*, 2019) daily e-mails in 2023. Spam has also grown gradually with the development of the internet and now 75% - 80% (Blanzieri and Bryl, 2008) of global email traffic is spam. Spam emails are useless, junk messages arrive into users' email inbox spontaneously and sent in bulk by the spammer. The main reason why people don't like spam is it's annoying to receive unknown and unwanted messages continuously during the day. Spam makes our email inboxes full instantly and reduces work efficiency and productivity. Users had to always keep sorting out legitimate messages from unwanted messages. Most spam messages have a commercial purpose. People who want to promote their commercial products send advertisements in bulk to spontaneous email IDs. Some malicious personals use spam messages to spread malware and links that lead to suspicious websites to gain private data of users.

A community is a social structure that shares personal values, cultural values, business goals, attitudes, or a world view. Frequently people in those communities work together to achieve important tasks. Those tasks can be varying from cleaning up the neighborhood to the takedown of governments and empires. An online community is connected by offering and accepting. Community is affinity, identity, and kinship that make room for ideas, thoughts, and solutions. In general, online communities value knowledge and information as currency or social resources. The difference between online communities from their physical counterparts is the extent and impact of "weak ties," which are the relationships acquaintances or strangers form to acquire information through online networks. Relationships among members in a virtual community tend to focus on information exchange, entertainment, professional, and sports virtual groups focused their activities on obtaining information(Horrigan, Rainie and S, 2001).

The aim of this project is to develop an anti-spam solution that will be based on community ratings to increase the accuracy of spam detection. Community ratings are very valuable. Because the ones who give feedbacks are humans. No algorithm can equate with a human who is using his previous experiences in the real-world to decide something. A large set of data given directly by users is always better than an algorithm that processes sample data. It will further enhance spam filters by providing real data from real human experience because spam filters always rely on predefined algorithms that work in the same manner repetitively. Community rating will fill a gap within automation. Spam filters can personalize themselves using given feedbacks to increase accuracy to reduce false positives and false negatives.

6. Scope

This project is a community-powered solution to spam filtering. It will consist of a spam filtering engine that will further enhance the accuracy of spam detection by using the input of user ratings. Often spam filters miss some of the spam messages and deliver them to inbox. In order to catch those spam messages, the system needs to increase the sensitivity of algorithms running inside the spam filters.

Usually, a community refers to a group of people living in a common geographical location. A certain community consists of thousands if not millions of people. People already do a lot of individual problem solving, and there's a good deal of merit in that. But many of the problems and challenges people face as individuals or as members belong to an organization affect everyone in the group. It makes sense then, that everyone being a contributor to a solution is very effective. Many things can be accomplished with a community full of people helping to solve a problem. The ability to collect such a large amount of data from a community without utilizing any extra effort is groundbreaking. People’s opinions about the experience they have with something is helpful information that can use to adjust that thing to fit their needs more accurately.

The spam filter will detect, and filter emails using spam filtering techniques and as a new technique, the system will request a rating from users to rate received spam emails missed by the spam filter. Users should consider the content of the email and decide If they are spam or not spam according to their own point of view. This technique will also help to personalize spam according to the user requirement. A spam email for someone may not be spam for another person. Therefore, individual feedback can also be used to meet individual needs while filtering messages.

7. Background

7.1. What is email?

Email also known as electronic mail or e-mail is a technique used to transfer messages between two electronic devices. More clearly, it is a text, photo, audio or video message transferring method between one or more individuals at once located in remote areas in different geographical locations via a computer network.

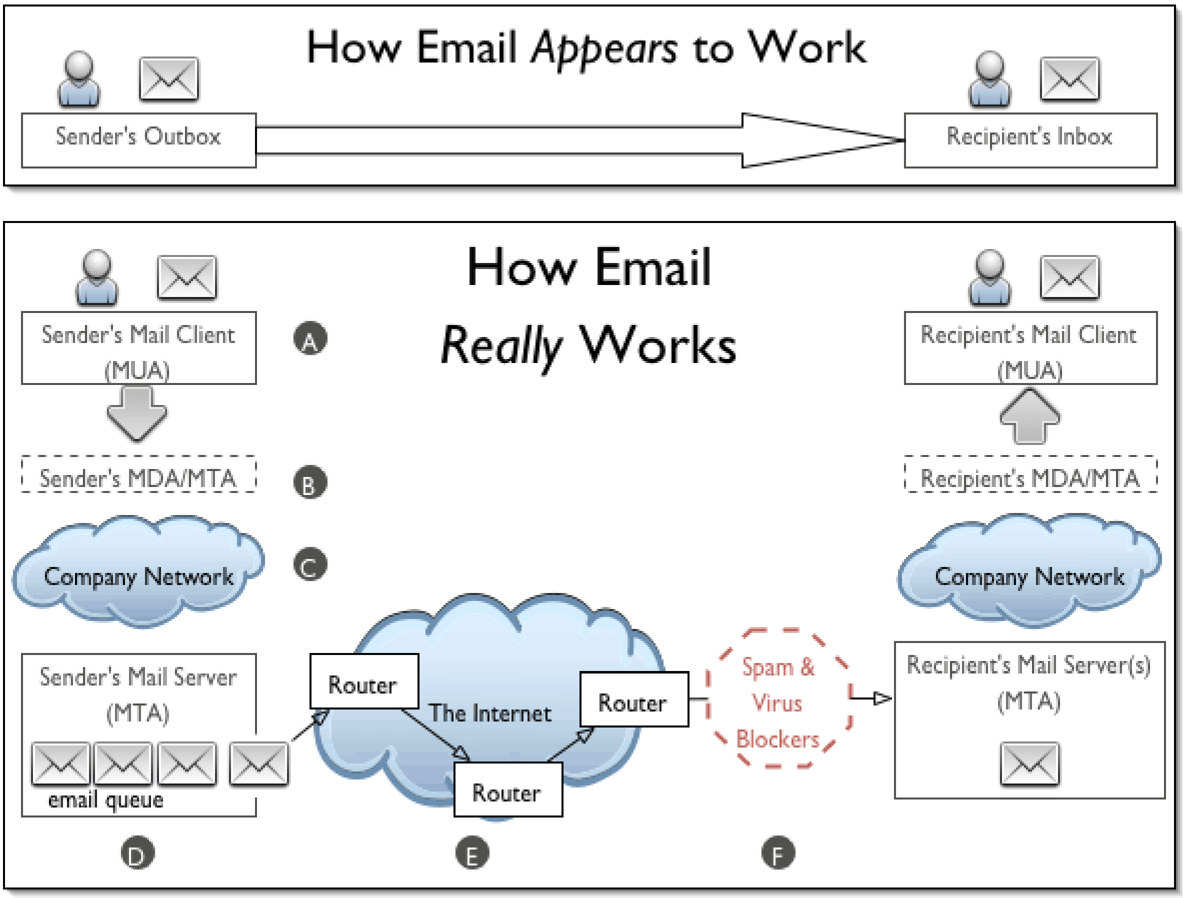


Figure 1: How email works
Image source: (*How Email Really Works*)

In 1971, Ray Tomlinson sent the world's first email to himself containing the message "something like QWERTYUIOP" using ARPANET. The Advanced Research Project Agency Network also known as ARPANET is the initial version of the modern internet used by the military of the United States of America. Email uses a centralized system which is also called an email server to transfer and store emails. Email server act as an intermediate between the sender and receiver of the email to facilitate and allow smooth transfer without any disruptions. In present, email sent by mobile holds 43 percent of e-mail opens (*Global email platform market share 2018 | Statista*, 2018). Desktop e-mail clients' open share had declined to 18 percent, and webmail accounted for 39 percent of opens. Based on the dominance of mobile, it is no surprise that the iPhone e-mail app was the most popular e-mail client, accounting for 29% of e-mail opens (*Global email platform market share 2018 | Statista*, 2018). Gmail was ranked second with a 27% open share. Gmail is a free e-mail service owned by Google, and the company reported 1.5 billion (*Number of Gmail active users 2018 | Statista*, 2018) active Gmail users worldwide in October 2018.

Email gives a user many advantages over traditional post mail. Anyone can send an email virtually without costing any money. The user only needs a working internet connection and an email account to send an email. Email can reach almost everywhere if that location has internet connectivity in an instant. Transferring multiple audios, videos, texts, and files is just a matter of user requirement. Environmental friendliness and ability for long-term storage are features totally ahead of traditional post mail.

7.3. What is email spam?

“Spam is the use of electronic messaging systems (including most broadcast media, digital delivery systems) to send unsolicited bulk messages indiscriminately” (Nazirova, 2011). History of email spam goes to the mid-1900s when the internet became a reality. In 1978, Gary Turk sent the first modern spam message to 400 people using ARPANET. Nowadays, everyone with an email account knows what is a spam message? Mail Abuse Prevention System (MAPS) is the first technology to defend against spam messages. It used blacklisting senders’ IP addresses to prevent future spam messages.

Spamming is a profitable business in today’s world. Most spammers spam to earn money by sending out advertisements to random email IDs to promote someone’s business products. The reason why there are so many spam messages is that it is so easy to create and spread. Anyone could easily become a spammer.

Spam now contains malware too. Once clicked by a user a malicious program will infect the system to steal private data or destroy information assets. Spam emails, when received in large numbers, overwhelm users’ mail inboxes, making it harder to find important messages and information. This leads to a decrease in overall productivity, efficiency, and loss of valuable time. More importantly, spam messages are the most common mediums used to perform phishing which is the initial step for executing large-scale cyberattacks. Phishing, if done successfully, can lead to devastating cyber threats like malware attacks and even ransomware attacks. Spam messages cause damages to the system and user at different levels. Spam consumes network bandwidth resulting in reduced work network efficiency and financial damage to the internet service provider or user. Even today for a normal user two or three spam messages received to the user inbox daily. Users should manually delete them. This causes a waste of time and a reduction in work productivity. Accidental loss of legitimate emails while deleting spam results in financial loss or time wastes to users or organizations. Some spam messages carrying malware may cause catastrophic damages to the whole IT system and organization if an employee opens that email. Sometimes the business may have to file for bankruptcy and face lawsuits if the company is regulated under the law of the European Union.

According to estimates by Alexander Ivanov, the President of the Russian Association of Networks and Services, three years ago Internet operators lost $55 million from the damage caused by spam. A person who reads 10-20 emails per day may receive in the region of 160-180 spam messages along with their business correspondence. That means that they will spend 5-6 hours per month just deleting spam, to the detriment of their productive working time ultimately causing financial damage. British Council, the United Kingdom's international organization for cultural relations and educational opportunities, has been hit by more than 10 million email attacks “in the lead up to Brexit” the organization has confirmed. According to Nimbus Hosting, which obtained the information via a Freedom of Information (FOI) request, the British Council blocked a total of 10,336,631 malicious emails last year. It also intercepted or blocked almost 200,000 emails that it suspected carried malware such as worms, Trojan horses or ransomware.

7.4. Types of spam emails

Spammers spam because their intended tasks still get fulfilled. People click on those spam emails and get caught by spammer’s trap or unintentionally fulfill what the spammer wanted. Even though only a small percentage of the population become victims of these spam attacks, In 2011, the total cost of USD 500 million financial loss suffered by victims of spam attacks worldwide (*Internet Crime Complaint Center (IC3) | IC3 2011 Annual Report on Internet Crime Released*, 2011). While the amount of spam received by a person rises, spammers use spam emails to fulfill different tasks. Spam emails can be divided into different categories according to their content. The following are ten types of spam emails that showed up regularly in the inbox.

7.4.1. Unsolicited advertisements

Unsolicited junk email advertisements are not addressed by name to an owner or any occupier of the premises. Food and clothing vouchers, magazines, leaflets from different institutions are the contents of these messages. Normally those emails are uninvited and from unknown sources. Mostly these types of emails are on the lower stack of email spam ladder. To avoid this kind of spam a user should avoid posting his/her email in public forums, blogs and avoid signing into unnecessary websites.

7.4.2. Phishing scams

This is the spamming type with the most risk of getting caught by spammers. Attackers create fake websites or announcements and deliver them to a unique community of people to trap them. Phishing emails are hard to distinguish from regular official emails. Most of the time spammers get successful in trapping people by phishing scams. Most financial losses due to spam emails have resulted from phishing scams. The main intention behind phishing is to gain usernames, passwords and credit card details from users. Then scammers will compromise user accounts on social media websites and blackmail the owner for financial gains or access online payments accounts like PayPal and withdraw all the money in the victim's account. If this kind of email appears always make sure to check the validity of the message from the official company that asked to provide this information. Phishing scams are very abundant and there is a very big probability of receiving such a scam to everyone who has an email account.

7.4.3. Nigerian 419 scams

Everyone has seen this type of spam message. These messages claim that the receiver has won a huge prize maybe millions of dollars from a competition or a lottery, in order to withdraw that cash prize, the victim should pay a commission or a small percentage of the prize as a service charge. There are different variations of stories scammers use. If a victim sends that small percentage of the cash prize to the scammer, basically victims lost their money, and nothing happens. Therefore, always avoid any emails claiming to be donating cash prizes.

7.4.4. Email spoofing

Email spoofing is using email addresses very similar to official addresses of well-known reputed companies to send scam messages appearing as a legitimate source. This trick is used to build trust and make users believable of fake sources and messages. The goal of those messages is to open and perform tasks asked by the message. Those messages are hard to distinguish from a genuine message.

7.4.5. Trojan horse email

Those emails contain malicious programs attached to them. Once clicked on the email attachments automatically get downloaded or ask the user to download and execute the program. They have malicious intent to damage information systems and electronic devices. When executed it will spread across the system and infect anything connected to the host device or system. Users need awareness to prevent those kinds of attacks. Even anti-virus software can't prevent infection methods the malware.

7.4.6. Commercial advertisements

This category includes spam emails sent by legit companies and websites with a commercial intention. When a user enters his/her email address to sign up or create an account on legit websites or companies, they use our email to send their commercial advertisements. Most websites ask permission from the user to send product updates, newsletters, and marketing emails. Some websites don't allow to create an account if a user doesn't agree to receive marketing emails. Therefore, these kinds of spam emails are different from previously mentioned unsolicited advertisements because the user agreed to receive them. One can use a secondary email address to sign up into those websites and avoid using the primary email address to prevent getting overwhelmed by commercial spam emails.

7.4.7. Anti-virus spam

Spam emails with convincing messages that indicate the user's computer or mobile device is infected with a virus include this category. Spammer asks the user to clean up the system as soon as possible using their anti-virus software. Software provided by the spammer may be a real malicious program. Once downloaded it will infect the system and may result in data theft, ransomware attack or system destruction. Installing a reputed and well known free or paid version of anti-virus can be helpful to solve the problem. When the user has an anti-virus program already installed there's no need for any other antivirus program to install when asked by a stranger.

7.4.8. Chain letters

Chain letters are messages that ask the recipient to forward the same message to many other people as much as possible. The spammer doesn't need to spend time collecting millions of email IDs to reach a bigger crowd. A spammer sends the message to a few people, and they will send it to as many as the possible multiplying number of recipients in each send. Normally these kinds of messages contain useless announcements but sometimes may contain hidden malicious programs. Chain letters are ideal to spread malware because tracing back to the originated source is virtually impossible and legitimate users contribute to spreading the malware unintentionally.

7.4.9. Political or terrorist pam

These spam emails claim to be sending from well-known politicians, political institutions, Government institutions. Political spam messages try to convince people about risk or threat to the users by appearing to be from government institutions and ask for private details or money to eliminate the risk while there's no actual threat that exists in the real world. Sometimes politicians use spammers to promote themselves and their political ideologies during election campaigns.

7.4.10.Porn spam

Pornography is a multi-billion dollar global business used by many people. Pornographic materials often use as click baits to get views to marketing websites and misleading digital materials. Porn spammers send pornographic materials to random email and often to people who had registered on pornographic websites using their emails. However, surfing porn on the internet and receiving porn spam emails is not related. It is true some of the time but not always. Normally email addresses are purchased by spammers from another spammer or website owners. If a user's email got purchased by a porn spammer that user will receive porn spam.

7.5. Spam filtering methods

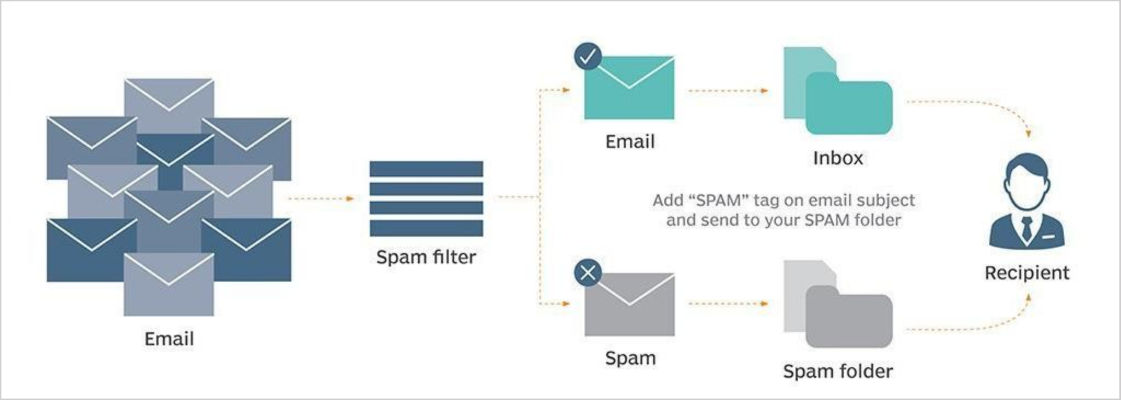

Email
Spam filter
Email
Inbox
Add "SPAM" tag on email subject and send to your SPAM folder
Recipient
Spam
Spam folder

Figure 2: How spam filtering works
Image source: (*What is email spam? - Definition from WhatIs.com*, 2017)

7.5.1. Filtering techniques

Spam filtering has come a long way in recent years, with complex algorithms identifying and catching spam before it becomes a risk to the business and users. Real emails can now pass safely through without ending up in the spam folder, and businesses can work with greater productivity and safety than ever before. People need email, but not spam or the damages it brings to businesses.

Anti-spam techniques also have been developed and used widely since the start of spam attacks in the early years. Yet existing techniques aren't 100% successful in blocking spam messages before arriving into email inbox but the rate of success has increased over the years. Anti-spam solutions automatically block those spam messages by analyzing the content, behavior, and sender of the message. Following are some of the techniques use to fight against spam regularly.

7.5.1.1. Content-based filtering

Content-based filtering uses words and phrases in an email message rather than using a predefined set of rules to decide the legitimacy of an email. This kind of spam filtering techniques mainly consists of complex algorithms constructed according to mathematical models. In order to find a solution to the problem of spam, the research community has undertaken a huge amount of work. Because machine learning approaches have succeeded in content filtering problems, these techniques have been adopted in spam filtering systems. Consequently, substantial work has been dedicated to the Naive Bayes classifier, with studies in anti-spam filtering confirming its effectiveness.

7.5.1.1.1. Word-based filtering

A word based spam filter is the simplest form of another generation of highly sophisticated spam filtering technologies. Those filters simply block messages that contain certain terms that may be related to spam. Normally, many spam messages have a commercial purpose they contain a specific set of words used to promoting as an example "free", "buy", "discount" etc. Sometimes legitimate messages may also contain those terms and spam filters may block them too resulting in increasing the number of false positives. Spammers often misspelled words to evade these kinds of spam filters. Therefore, continuously updating the word list is important.

7.5.1.1.2. Heuristic filters

The heuristic filter is an extended version of a word-based filter that relies on predefined rules to check the validity of the email. The heuristic filter checks the contents of spam as multiple terms and phrases increasing the success rate of detection. It rates words according to the frequency and probability of appearing in spam emails. The total score achieved by the email decides the email being spam or not spam. The cutoff score is decided by the system administrator. Heuristic filters are effective when they are configured correctly then the techniques mentioned previously.

7.5.1.1.3. Signature matching

Many anti-spam and anti-virus software vendors use this method to detect spam. An organization that uses signature matching to spam detection maintains a set of email accounts from different service providers. When a spam message is received to one of that account a unique signature is generated to that spam message and distributed stored in a centralized database. Then customers can update their client software to get to know of new spam signatures and block them when the spam arrives by comparing the signature to a spam message in their inbox. When a user receives a new spam message from an unknown source it also generates a signature and sends it to the central database. This technology uses low resources to perform tasks and the false-positive rate is very low. Because the signatures are unique spam and legitimate messages won’t have the same signature. Signature matching is very vulnerable to attacks. It has a single point of failure due to its single central database. If the central database is compromised the intended task of delivering new signatures won’t be completed.

7.5.1.1.4. Bayesian filters

Bayesian filters are considered the most advanced, sophisticated and highly effective spam filters of all time. These filters use a mathematical model to calculate the probability of the content of the email being spam. When used over a period it becomes highly effective by learning from previous email messages which are spam and not spam. It uses previous data to decide the validity of future messages. To detect which one is a spam message it compares the content of the message to previous data to calculate the probability of being spam or not spam. The main advantage of the Bayesian filter is it required little or no maintenance. It will automatically collect the data needed for processing.

7.5.1.2. Header based filtering

An email message typically consists of header and body. The header is a necessary component of any email message. The Simple Mail Transfer Protocol (SMTP) defines a set of fields to be contained in the email message header to achieve the successful delivery of email messages and to provide important information for the recipient. These fields include email history, email date, time, the sender of the email, receivers of the email, email ID, email subject, IP addresses, etc. Header-based email spam filtering represents an efficient and lightweight approach to achieve the filtering of spam messages by inspecting email message header information. Mainly sender of the email which is the originated source and IP address of the source in the header is used to decide whether the email is spam or not spam. Extracted IP addresses and sender of the email will be compared to a list of preoccupied malicious IP addresses and senders.

7.5.1.2.1. Blacklisting

This is the most primitive and oldest technique used to tackle spam emails from the early 1900s when the internet invented, and spam started to grow. Basically, what happens is totally blocking any IP addresses that send spam messages. It will eliminate the ability to contact the user once to get blocked. Next time when the spam emails arrive system will compare its IP address to the blacklist maintained by the system administrator. If the IP address matches connectivity will be terminated and email is rejected. One downside of blacklisting is it may block legitimate users due to any misidentification resulting in more issues. Blocking legitimate users is more damaging and costly than receiving spam emails. Spammers frequently change IP addresses, therefore, blocking them every time when they come up with a new way is time consuming and not efficient as it builds up huge lists of IP addresses and email IDs. In this age blacklisting alone is far from an ideal anti-spam solution.

7.5.1.2.2. Real-time blackhole list

This type is a new generation of traditional backlisting. It needs less hand-on work to maintain the blacklist. Normally, real-time blackhole lists are maintained by third parties. A user can subscribe to a blacklisting service to get the filtering done. Once subscribed every email a user receives will be sent through the third party service for the filtration process before received by the user. Still, there's a risk of blocking legitimate users but the risk is a little bit high because the service provider is a third party and the organization or service receiver doesn't have control over that.

7.5.1.2.3. Whitelisting

As the name reveals Whitelisting is totally opposite of blacklisting. The administrator maintains a list of IP addresses that only allow email communication while blacklist block the ability of IP addresses in the list to communicate. A whitelist is a list of trusted-users. The disadvantage of whitelist is unknown legitimate users won't be allowed to communicate. Some whitelisting solutions check the IP address of a new sender with a database to find out any history of spamming and it will be whitelisted if there's no spamming history. As same to blacklisting, whitelisting alone also not ideal to solution for spam filtering.

7.5.1.2.4. Greylisting

Greylisting is a new technique. Greylisting assumes spammers only send a bulk of spam once. They don't bother about whether the email is received or not by the other end because they send the same message to millions of email addresses, missing out few is really doesn't matter. Legitimate servers always try a few times to send if failed, therefore distinguishing can be easily done between the two. Greylisting only allows a sender to communicate if it passes this “try again” test. If passes sender will be treated as a legitimate user. Greylisting consumes more system resources and delays message delivery which is a problem if the communicators' purpose is time-sensitive.

7.5.1.3. Source-based filtering

An SMTP (Simple Mail Transfer Protocol) server is an application that’s primary purpose is to send, receive, and/or relay outgoing mail between email senders and receivers. In order to send an email, every SMTP server must be registered as a legitimated email sender to deliver messages successfully. An SMTP server also has one or many domains. Spammers use a fake or unauthentic server or domain to create spam attacks. Source-based filtering mainly looks at the source of the origin of the email. Often seek the validity of domain or server of origin.

7.5.1.3.1. DNS lookup systems

DNS lookup is not a very reliable and efficient method alone. This checks the domain of an email address to validate whether it sent by a real mail server or not. It produces many false positives. Because spammers frequently change email IDs, IP addresses and domains. Once detected as spam that IP address will be blacklisted and blocked. Sometimes spammers can use proxies to hide their real IP addresses to trick the DNS lookup and bypass the security mechanism.

7.5.1.3.2. Challenge/Response system

These systems challenge the sender of the message to get a response within a certain time period. As an example, a challenge/response system asks the sender to complete a certain task to sometimes enter a code or complete a puzzle to check whether the sender is legitimate. Because a spammer who sends millions of spam messages won’t respond to these challenges and the challenge provided by the system can only be completed by a human. A server or bot system won’t be sophisticated to complete such a task. In real-world people aren’t interested in such time-consuming tasks just to deliver a legitimate email. Therefore, it is not that suitable to use as an everyday spam filtering technique except in special situations.

7.5.2. Filtering approach

7.5.2.1. Server-side filtering

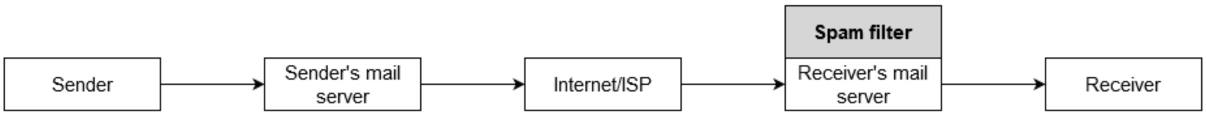


Figure 3: Server-side filter

Server-side Junk Mail Filters can certainly be viewed as a specific type of e-mail program which would allow e-mail subscribers to disregard unwanted e-mail messages. This particular sort of e-mail tool functions in accordance with a set of user-defined filter rules on the server-side. The benefit of establishing a server-side rule would be that the user does not be required to have an email client attached to the mailbox for the spam filtering rule to operate. Whenever the email comes, it will immediately be redirected to the correct folder based on the way the rule is configured. The downside of this setup would be that it will direct the message to a Junk E-Mail folder in the mailbox on the server. When working with the POP3 protocol to connect with the mailbox, the computer user will not have access to it unless log into the Web client App to go through the contents of the Junk E-Mail folder.

7.5.2.2. Client-side filtering

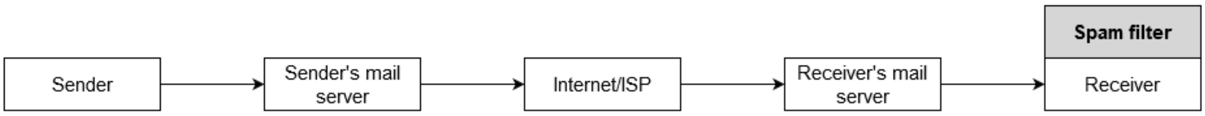


Figure 4: Client-side filter

Client-side spam filtering software resides on the user 's private computer, performing spam filtering as receive it. This kind of software program might be readily available as a stand-alone system which filters mail just before reaching the email client or perhaps as an add-on combines with the e-mail client. Stand-alone software program grants the independence to keep on utilizing the e-mail client of the user 's choice, while an add-on would require utilizing an email client which is actually reliant on the software application. Nonetheless, add-on anti-spam utilities typically demand less effort with regards to controlling spam by providing management possibilities to the email client itself. When deployed throughout a network, client-side antispam software application allows every end user to configure the software application so that it performs in accordance with that user 's individual desires. But system administrators are going to have a rough time enforcing and implementing spam as well as email utilization policies across a network due to its decentralized approach. Clients with unique filtering necessity as a good example somebody who wishes all spam with the term "news" to nevertheless go to their Inbox, but does not want to see some other spam, can certainly turn off server-side filtering, and create their own filter rules for the header in the email client application. This is referred to as client-side filtering". Doing this is much more complex and also less efficient.

7.5.2.3. ISP-level filtering

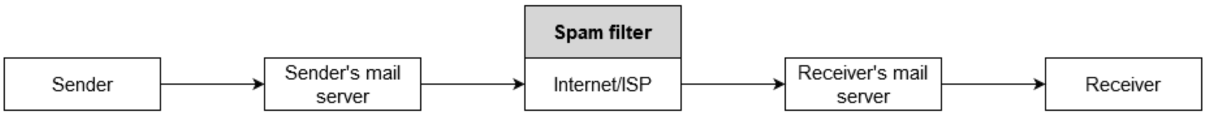


Figure 5: ISP-level filter

Historically an Internet Service Provider (ISP) was an entity that provided a connection to the internet for its customers, such as Dialog or SLT. Now ISPs are anyone who provides an email address, like Outlook, Gmail, Yahoo, and AOL. ISPs use a variety of tools to process and filter incoming email. ISPs use spam-blocking measures to help sort through incoming email and identify the spam to keep their own customers happy. Once they've identified a message as unwanted or as spam, ISPs do their best to block all similar messages; they have no way of knowing if the messages were requested in the first place. Sometimes these measures can inadvertently block the delivery of legitimate emails; when this happens, it's called a "false positive." ISPs use reputation systems to filter mail, taking data from many external sources as well as their own internal data, to determine if an email should be considered spam or not spam. Reputation systems are more robust than traditional content filtering and email blocklists that often use a single bit of content or images, and instead, use many data points to score incoming mail.

7.6. Email management and security solutions

7.6.1. Spam gateways (Spam filters)

7.6.1.1. MailCleaner

Figure 6: MailCleaner logo
Image source: (*mail cleaner logo - Google Search*)

MailCleaner is a really powerful antivirus and anti-spam system. Based on the latest generation of filtering technology, MailCleaner doesn't need to be placed on user 's computer. Rather, it acts before e-mail reach user 's mailbox, at probably the highest level of the networking infrastructure of company, organization or ISP. MailCleaner relies on complex rules which are actually updated each day by the people of the MailCleaner Researching Center against to spammers' continously changing techniques in creating new viruses (*Documentation - MailCleaner - Open source Anti spam & Antivirus gateway*, no date).

7.6.1.2. SpamAssassin

Figure 7: Apache SpamAssassin logo
Image source: (*Apache SpamAssassin: Welcome*)

Apache SpamAssassin is actually the #1 Open Source anti-spam platform giving administrators a filter to classify email and block "spam" (unsolicited bulk email). It uses a sturdy scoring plug-ins and framework to incorporate a broad range of innovative heuristic as well as statistical analysis assessments on email headers along with body text like text analysis, Bayesian filtering, DNS blocklists, and collaborative filtering sources.

Apache SpamAssassin generally differentiates successfully between ham and spam in between 95% to 100% of cases, based on what type of mail you receive and your instruction of its Bayesian filtration system. Specifically, Apache SpamAssassin happens to be shown to generate around 1.5 % false negatives (spam which was missed) and around 0.06 % false positives (ham improperly marked as spam) (*SpamAssassin: Documentation*, no date).

7.6.1.3. eFa Project

Figure 8: eFa project logo
Image source: (*eFa project – email FILTER appliance project*)

eFa means Email Filter Appliance The concept about eFa is actually creating a (virtual) system to overcome spam utilizing opensource tools. eFa is actually striving to put together existing open-source projects into one ready to use and quick to deploy package. eFa is born out of requirement for a cost-effective email virus & spam scanning alternative after the ESVA died. Consequently, in 2012 the eFa undertaking began, the original idea was using a totally different setup than ESVA, that resulted in eFa 0.2 as the very first public release. eFa 0.2 was founded on Debian for the Operating system, Exim for the Mail Transport Agent and Baruwa as the web graphical user interface, several extra features had been added in eFa 0.3 that was released in January 2013 (*efa_project_v4 [eFa project Wiki]*, no date).

7.6.1.4. MailScanner

Figure 9: MailScanner logo
Image source: (*Home - MailScanner*)

MailScanner is an incredibly recognized open source email security system. It is employed at more than 30,000 websites around the globe, protecting leading government departments, educational institutions and commercial corporations. This particular technological innovation is fast becoming the conventional email filtering alternative at numerous ISPs for virus protection as well as spam filtering. MailScanner scans all e-mail for viruses, spam and attacks to protect against security vulnerabilities as well as plays a significant aspect in the protection of a network. To properly accomplish this function, it have to be trustworthy and reliable. The sole method to attain the necessary level of trust is usually to be open source, an approach the industrial {suppliers|vendors} are unwilling to take. By virtue of simply being open source, the technologies in MailScanner has been reviewed more often than not over by several of the brightest and best in the discipline of computer security and safety, from across the globe. MailScanner has been invented by Julian Field at a world-leading Electronics and Computer Science Department at the University of Southampton (J. Field, no date).

7.6.1.5. OrangeAssassin

Figure 10: OrangeAssasin logo
Image source: (*orangeassassin logo - Google Search*)

OrangeAssassin was designed as an enhanced open source drop in substitute for SpamAssassin for Linux computer users as well as control panel providers. Many standard SpamAssassin plugins are actually offered. Furthermore, the framework supports effortless development of completely new plugins. OrangeAssassin comes with thorough updated documentation regarding compatibilities, licensing and configurations. The source code is actually open and also makes it possible for forking and submission of pull requests. OrangeAssassin is actually appropriate for Python 2.7, Python 3.2 and later, PyPy3 and PyPy. OrangeAssassin delivers resource consumption optimisation (memory, CPU, disk IO) (*OrangeAssassin | New Open Source Antispam Email Filtering Framework*, no date).

7.6.2. SMTP servers (Mail servers)

7.6.2.1. Zimbra

Figure 11: Zimbra logo
Image source: (*The World's Leading Email Collaboration Platform - Zimbra*)

Zimbra Collaboration is a full featured messaging as well as cooperation solution which usually comes with Web document, tasks, calendaring, address book, and email authoring. Wherever possible, Zimbra Collaboration adheres to present industry standards and open source implementations for backup handling, operating platform, user authentication, and database management. Zimbra is actually an enterprise class email, calendar and also cooperation solution designed for the cloud, all public & private. With a redesigned browser-based interface, Zimbra offers probably the most impressive messaging sensation we have today, joining end users to the information and actions in the personal clouds of theirs. Zimbra is viewed as a substitute to Microsoft Exchange Server (*Best Open Source E-Mail-Server*, no date).

7.6.2.2. iRedmail

Figure 12: iRedMail logo
Image source: (*iredmail logo - Google Search*)

iRedMail is actually a full fledged, flexible and stable Linux Mail Server dependent on Dovecot IMAP server and Postfix Mail agent. Employing iRedMail you are able to very easily deploy a Linux Mail Server in couple of minutes. One of the primary positive aspects with iRedMail is the fact that it will immediately setup all essential elements with a lot less user 's interaction.

7.6.2.3. SquirrelMail

Figure 13: SquirrelMail logo
Image source: (*squirrelmail logo - Google Search*)

SquirrelMail is actually a standards based webmail program developed in PHP. It consists of built in clean PHP support for the SMTP and IMAP protocols, and all webpages render in clean HTML 4.0 (with absolutely no JavaScript required) for optimum compatibility across browsers. It has very few needs and is also extremely convenient to configure and also install. SquirrelMail has all the functionality you would want from an email client, including strong MIME support, address books, and folder manipulation (*SquirrelMail - Webmail for Nuts!*, no date).

7.6.2.4. Sendmail

Figure 14: SendMail logo

Image source: (*sendmail logo - Google Search*)

Sendmail implements a standard purpose internetwork mail routing facility underneath the UNIX© operating system. It is not linked to any kind of one particular transport protocol -- could be compared to a crossbar switch, relaying communications from a single domain into a different. In the procedure, it is able to do a restricted quantity of message header editing to place the message into a format that is actually suitable for the acquiring domain. Every one of this is accomplished within the regulation of a configuration file. Mainly because of the needs of versatility for sendmail, the configuration file is able to seem somewhat unapproachable. Nevertheless, there are just a couple of fundamental configurations for the majority of websites, for which basic configuration data have been supplied. A good number of alternative configurations could be created by modifying a current configuration file incrementally (Eric Allman, no date).

7.6.2.5. Postfix

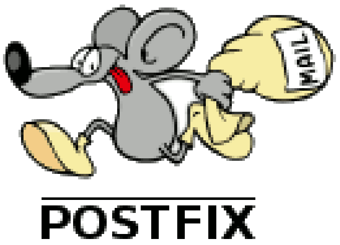


Figure 15: Postfix logo
Image source: (*postfix logo - Google Search*)

Postfix is actually an open-source and free mail transfer agent (MTA) which routes as well as delivers electronic mail. It is released under the IBM Public License 1.0 which in turn is actually a totally free software program license. On the other hand, beginning with edition 3.2.5, it is readily available under the Eclipse Public License 2.0 at the user 's choice. Initially written in 1997 by Wietse Venema at the IBM Thomas J. Watson Research Center in New York, and originally launched in December 1998, Postfix remains as of 2020 to be actively developed by the creator of its as well as other contributors. The application is also recognized by its former names IBM and VMailer Secure Mailer (*Postfix (software) - Wikipedia*, no date).

8. Literature review

In September 2019, spam emails accounted for 54.68% of global email traffic (*Spam statistics: spam e-mail traffic share 2019 | Statista*, 2019). In that year China is responsible for generating 20.43% of global spam traffic while the United States is responsible for 13.37% (*Spam e-mail: countries of origin 2019 | Statista*, 2019). Global annual spam traffic was 69% in 2012 and it had decreased to 55% in 2018 and 53.68% in 2019 (*E-mail spam rate worldwide 2018 | Statista*, 2018). In 2018, the Federal Bureau of Investigation (FBI) of the United States stated in a report that financial loss due to business email compromise and email spam is USD 12.5 billion worldwide (*Internet Crime Complaint Center (IC3) | Business E-mail Compromise The 12 Billion Dollar Scam*, 2018).

Current defense strategies in fighting email spam are much more sophisticated than a decade ago. The research focused on old everyday anti-spam techniques (Christina, Karpagavalli and Suganya, 2010) and modern approaches like machine learning (Bhowmick and Hazarika, 2016) have improved success rates of spam detection to virtually 100%. Old rule-based methods easy to deploy, configure and hard to maintain are becoming less efficient and producing more false positives. Modern machine learning techniques based on self-learning mathematical models are easy to deploy and need less maintenance to produce much less false positives and have a higher success rate of spam detection. Machine learning models learn to adapt themselves to new situations and needs fewer hands-on configurations. Thus, those techniques need heavy computing power to process algorithms and come up with results. Because these algorithms success rate based on previous data it needs to store more data consuming more storage. Statistical spam filters like Naïve Bayes, Term Frequency-Inverse Document Frequency, K-Nearest Neighbor, Support Vector Machine, and Bayes Additive Regression Tree are being used more widely these days. Out of them, the Naïve Bayes algorithm is the most prominent and has a higher success rate (Banday and Jan, 2009). Neural networks are the next possible technology that will enable more success rates and low false negatives in spam detection. This technology imitates the human brain structure to process data. Neural networks try to understand the underlying relationships between data.

There are many mail servers present in today. Among them there are very few commercial mail servers like Microsoft Exchange server and Axigen, Open source mail servers like Postfix, Zimbra, Apache James, Squirrelmail, Sendmail and iRedmail dominate the market. In a survey conducted in 2018 found out that, Exim, Postfix, Sendmail, MailEnable, MDaemon and Microsoft are the top mail servers in use, representing 56.78%, 33.77%, 4.43%, 2.22%, 1.05% and 0.8% of mail servers worldwide respectively (*Mail (MX) Server Survey*, 2018). Spam Titan, Vircom, Spambrella, Comodo are top commercial spam filtering service providers and MailCleaner, SpamAssassin, OrangeAssasin, Rspamd, eFa and ASSP are open source solutions available. In this project, the OrangeAssasin spam filter is being used. OrangeAssasin is an open-source python-based drop-in replacement for the SpamAssassin spam filter. As the mail server, Postfix Open source mail server is used. It is light weight and easy to configure mail server that is being used worldwide by IT professionals.

9. Business case

Recently, an email scam was carried out by a person in France impersonating former minister of defense, France. The attacker has used legitimate-looking email IDs, letterheads and information to communicate through email. The attacker has scammed about 150 different high government officials, diplomats, businessmen, and ministers of other countries and gained 50 million Euros. All of that was done just using emails and short video calls (*Defense Minister Was on the Line, Asking for Millions to Aid France. Or Was He? - The New York Times*, 2020). The University of Plymouth which I'm currently studying is also continuously being affected by phishing scam emails. Every year students and staff receive at least ten phishing emails claiming to be from university administration mentioning requests for wire payments, free gifts, surveys, virus infections, refunds, attempting to obtain email credentials etc. (*Phishing Line - IT Strategy & Architecture*, no date). Likewise, businesses and organizations worldwide are being targeted by malicious attackers. An estimated amount of 45 billion US dollars have been the cost of damage caused by cyberattacks worldwide in 2018 as businesses struggle to cope with ransomware and other malicious attacks (*Cyber Attacks Cost $45 billion in 2018 | 2019-07-10 | Security Magazine*, 2019).

Spam is also a productivity drainer. Employees can waste a considerable amount of time dealing with unsolicited and unwanted emails. Even five minutes a day spent dealing with spam emails means major productivity losses for employers. Five minutes lost each day by 100 employees adds up to a full eight hours. Multiply that by 250 working days a year and over the course of 12 months, more than 52 days of work will be lost. A spam filtering solution cannot be 100 percent effective. However, a business email system without spam filtering is highly vulnerable, if not unusable. It is important to stop as much spam as you can, to protect your network from the many possible risks such as viruses, phishing attacks, compromised web links and other malicious content. Spam filters also protect your servers from being overloaded with non-essential emails, and the worse problem of being infected with spam software that may turn them into spam servers themselves. By preventing spam email from reaching your employees' mailboxes, spam filters give an additional layer of protection to your users, your network, and your business.

Anti-spam filters automatically quarantine the spam emails, ensuring the inbox is spam-free. Quarantined emails are kept for a fixed number of days and then discarded. During that period, you can check and recover any legitimate email that may have been quarantined. Most of the antivirus software comes with an automatic filter update feature for the timely detection of new types of Malware threats. Automatic updates not only help the anti-spam software to stay up-to-date, but it also helps secure your system from new kinds of Malware. Monitoring techniques can help to monitor and filter out spam from multiple accounts. You can filter your home email from work email, and vice versa. Anti-spam software allows you to maintain a trusted list of people whose emails you wish to accept. These emails will never be mistaken for spam as against the blacklist of spammers. You can also update the list in the future. Some anti-spam software allows you to report spam back to the company supplying the software. It helps that company to develop a new type of filters based on the analysis of the reported spam.

EveryCloud, SpamTitan, Vircom, Spambrella, Cyren, Securence, Comodo, Proofpoint, AppRiver, Mimecast and MailChannels are the best existing companies which supply best antispam solutions. Some of the services they provide include Email antispam and antivirus, archiving, encryption. Web filtering, outbound mail filtering, IT managed services, sandboxing, digital certification and threat intelligence (*The Top Spam Filtering Companies - February 2020 | 99firms*, 2020). Using an antispam solution in corporations is critical due to the above mentioned threats. When an antispam solution is used increased productivity and email management, risk of being a victim to phishing attacks and scams, saving network and storage resources and eventually all these benefits leading to a reduction of financial loss.

10. Motivation

Currently, organizations implement their own infrastructure and techniques to gain more control over information assets and to achieve better performance and improved data security. In the present time, Email which is such an information asset is the main method used to conduct formal communication. Like other technologies email also has both advantages and disadvantages. Therefore, virtually every organization use their own email services. In such cases, Email spam is a big disadvantage. Spam is present almost everywhere. Every person's inbox has spam messages. Present antispam technologies are more sophisticated than ever before yet can't achieve a 100 percent success rate in spam detection. At least 1 percent of spam messages bypass spam filters into email inboxes. When spam has arrived in the email inbox a user can see the spam message.

Starting from 7 million years ago humans had evolved as individuals as well as community groups. A single individual by himself is useless to him and to the group or his community unless he or she is the leader or king of the group. Humans have achieved tasks as groups. They waged wars, won wars, went on hunting, did religious activities, built cities, kingdoms, empires and wiped out entire civilizations. Community is always a part of human lives because humans are social animals. Consequently, people used ideas, opinions and advices of a elders, ancestors or well informed group of people like religious leaders, scientists before taking decisions or doing something. Throughout the human history that's what people did and still a community is a very powerful entity.

Therefore, creating a method to get data about spam messages is easy because we can directly ask from the user themselves. Which is the community of people who owns an email account. As we know the internet is growing day by day from being a small network to large scale enterprise networks. Every individual and organization tries to keep up to date with forever innovating and upgrading technologies. In 2019 there were 3.8 billion active email accounts in the world, and it predicted to be rise to 4.4 billion in 2023 (*Number of e-mail users worldwide 2023 | Statista*, 2019). If we can get feedback from at least 1% of 4.4 Billion users which is 44 million users is still a very significant amount. When the such large number of users gave their feedback about spam messages in the inbox that information can be used to improve the accuracy of future spam detection by running through an algorithm. A community feedback method which is a more sophisticated technique allows users to help the organization to identify spam by providing their feedback. It also helps to customize emails received by individual users. By adopting a defense in more depth, it is possible to block new varieties of malware, spear-phishing attempts, and zero-day attacks and ensure that these sophisticated new threats do not get delivered to users' mailboxes.

11. Approach

Institutes and personnel responsible and interested in developing and innovating new technologies had invented new ways and techniques to fight against spam messages. Spam filters are deploying on server-side and client-side to detect spam. Later server-side approach became more popular. Because client-side processing is not efficient and successful when the internet started to grow, and the number of spam messages increased day by day. Client-side spam filtering doesn't address the very aim of spam filtering, which is stopping spam from arriving into the client-side or user inbox. Server-side spam filtering is very efficient and productive. When deployed on the server-side there are abundant resources in servers that can be used to data processing and it also prevents spam from being delivered to the client-side. It avoids resource wastage like bandwidth consumption and saves the time of users. A server-side spam filter detects and blocks spam at the server level.

Server-side spam filtering can be achieved using a central spam filter located before the mail server. Which is also known as spam filtering gateway. An antispam gateway inputs all emails incoming from different email routes before entering to internal mail server. Every email must pass through antispam gateway before getting delivered to mail server. After filtration process, the remains which are the legitimate emails will be forwarded to internal mail server. Mail server will manage and deliver emails to clients.

When detecting spam various methods are being used. Due to the advancement in technologies filters become more and more sophisticated. Machine learning and artificial intelligence are becoming the trend in surpassing best existing antispam techniques like Naïve Bayes filter which use statistical analysis. Another method which is very productive is crowd sourcing the spam filtering problem. As people are becoming more educated about spams, SMS providers, Email providers and social network platforms are using crowdsourcing to combat spam messages. By soliciting contributions from groups of users may significantly improve the effectiveness underlying spam filtering mechanism (*Future trends in spam filtering | datascienceCMU*, 2014). Artificial intelligence can be used to analyze situations that humans can't even imagine doing in real time. Data from different sources like social media communications, forums, chats, news articles can be analyzed to detect any triggering situations in no time. Around the world normally 75% of the emails that people report as spam are legitimate newsletters (*Can artificial intelligence spot spam quicker than humans?*, 2019). This makes a grey zone in spam filtering. In such situation identification of an email as spam or ham become difficult. Artificial intelligence can be used in such situations effectively. Overtime spam filter will also learn to fulfil individual needs based on data of individual reporting statistics.

12. Method

This project consists of two main parts which are spam engine and email rating system. This whole development process involves seven stages. The stages of this project are background research, project initiation, high level design, setting up the environment, development of spam engine, development of email rating system, testing and evaluation and finalizing.

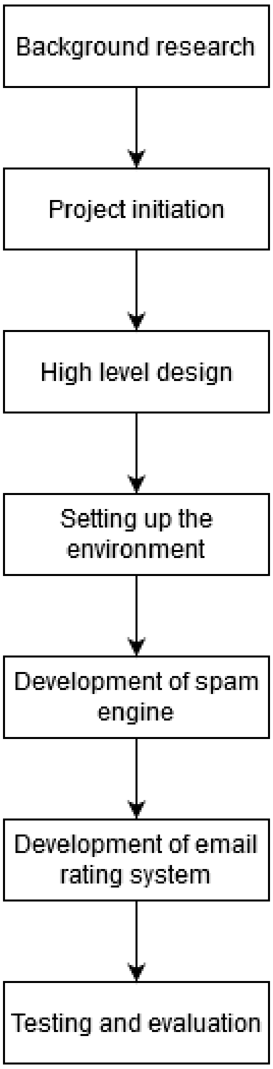

Background research
Project initiation
High level design
Setting up the environment
Development of spam engine
Development of email rating system
Testing and evaluation

Figure 16: Project methodology

13.1 Background research

Starting from the background research and literature review which allowed to do a comprehensive analysis of existing technologies, tools, techniques and methods of spam filtering, computer programming and computer networking. Background research establishes the context of the research and project. Knowledge was acquired by reading blog articles, reading tutoring websites like www.tutorialspoint.com, www.khanacademy.org and www.geek-university.com, watching YouTube tutorials, reading research papers from websites like www.scholar.google.com, www.elsevier.com, www.sciencedirect.com and having discussions and meetings with project supervisor. First meeting with project supervisor was carried out and discussed about project proposal and existing products related to my project. Background research helped to define a more solid project scope by understanding unclear issues and thoughts regarding the project. Well defined scope helped to carry out more specific and relevant project tasks easily.

13.2 Project initiation

In the project initiation stage project initiation document was prepared and submitted. Second supervisory meeting with supervisor was carried out in order to discuss about project more in depth. At the meeting existing technologies and tools, corrections to project scope, what are the technologies I can used to do this project were discussed. Advices on how to write introduction chapter, scope chapter, background chapter, structure and outline of final report were given.

13.3 High level design

In the high level design, all the abstract planning and designs were done. Among them were database design, interface design, system architecture etc. Those designs helped in understanding more about what the project is going to be and the outcome.

13.3.1 Plugin interface

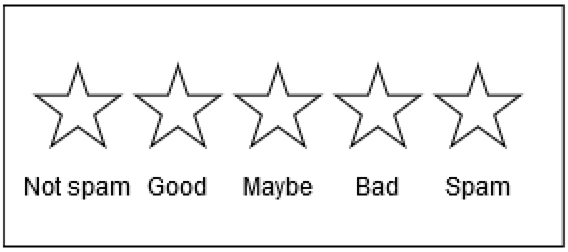


Figure 18: Extension before getting ratings

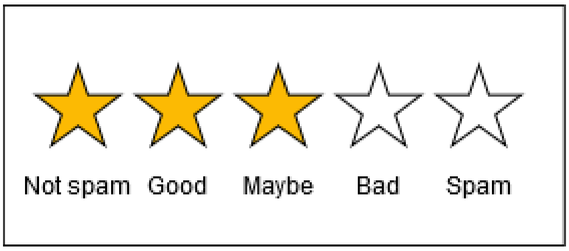


Figure 17: Extension after getting ratings

13.3.2 System design I

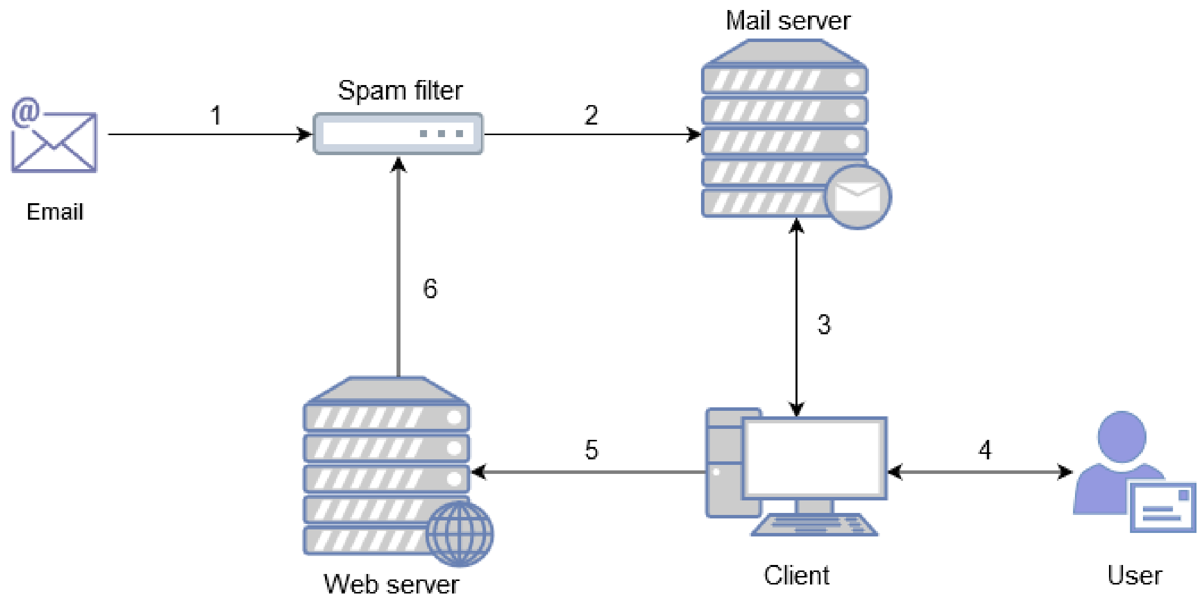


Figure 19: System design I

- ➢ 1 - Incoming emails to spam filter from internet
- ➢ 2 - Filtered emails sent to mail server
- ➢ 3 - Client machine/application retrieve emails in the mail server
- ➢ 4 - User watch the emails and give feedback
- ➢ 5 - Send user feedbacks to web server for further evaluation
- ➢ 6 - Web server sent spam updates to spam filter

13.3.3 System design II

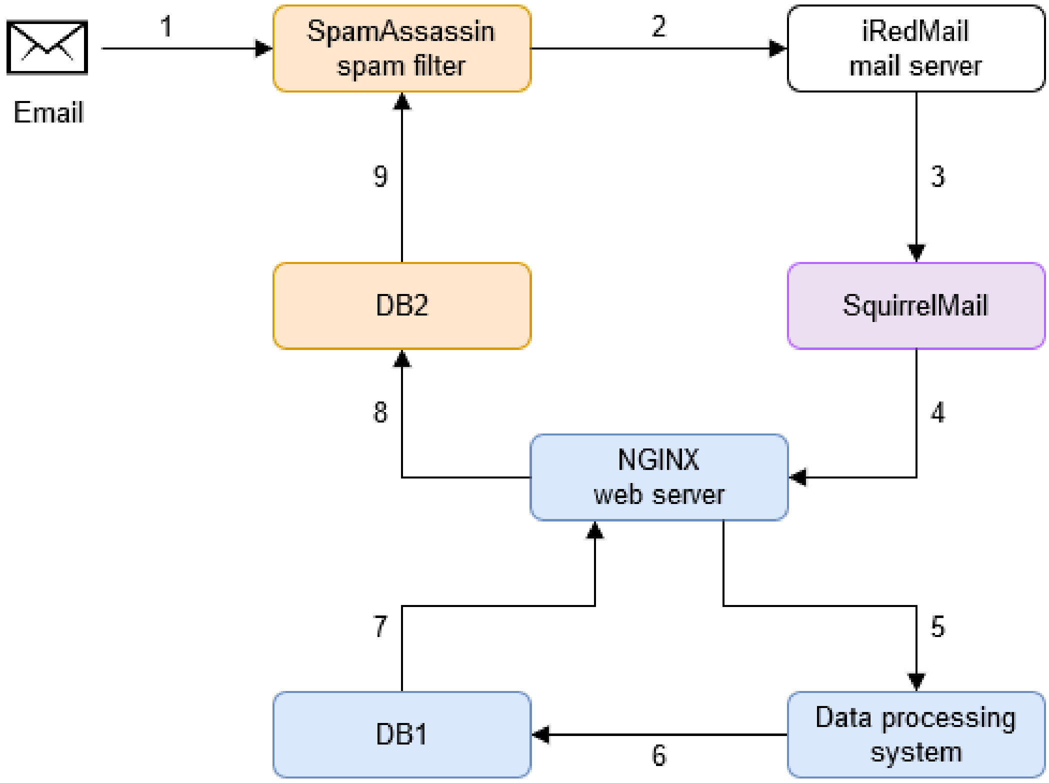


Figure 20: System design II

- ➢ 1 – Incoming emails to SpamAssassin spam filter from internet
- ➢ 2 – Filtered emails sent to iRedMail mail server
- ➢ 3 – Retrieve emails from SquirrelMail email client
- ➢ 4 – Send feedback to Nginx web server via email rating extension
- ➢ 5 – NGINX web server sends feedbacks to data processing system in server
- ➢ 6 – Data processing system send processed data about spam to database 1
- ➢ 7 – NGINX web server retrieves spam records from database 1 (DB1) in server
- ➢ 8 – NGINX web server sends spam records to database 2(DB2) in spam filter
- ➢ 9 – Spam filter retrieve spam updates from database 2(DB2)

13.3.4 Spam filtering process

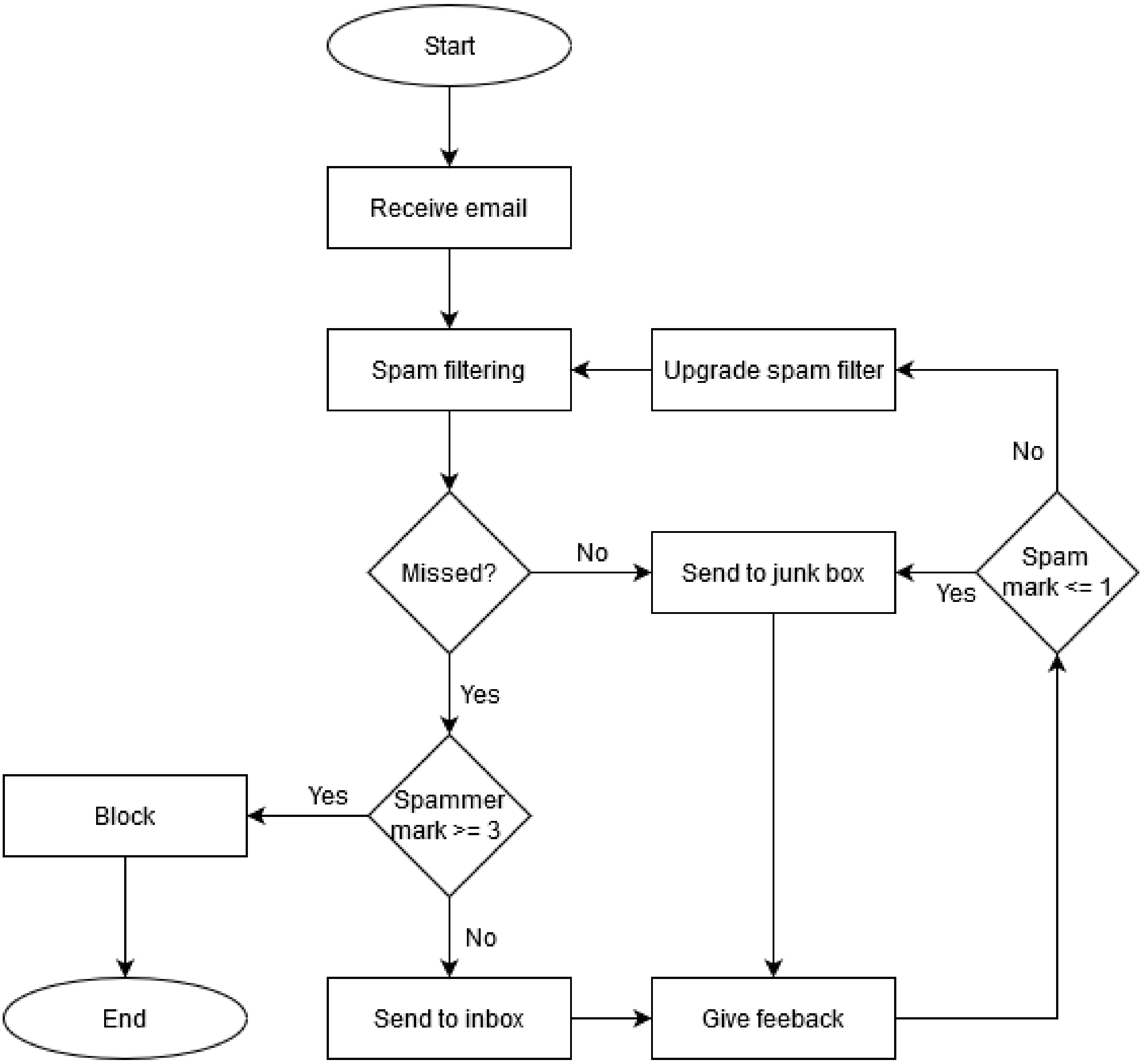


Figure 21: Spam filtering process

13.4 Setting up the environment

For this system to work main requirement is an email server. Therefore, CentOS was installed which is a free Linux operating system in VMware player for this purpose. Inside the virtual machine iRedMail email server is installed which runs in localhost. iRedmail server comprises of Postfix SMTP server and Dovecot IMAP server. SquirrelMail is used as the email client. SquirrelMail is a web application. It also runs on NGINX web server in the localhost. MariaDB with MySQL is used as the database software.

13.5 Development of email rating system

Developing an email rating subsystem, which is the SquirrelMail plugin. A widget or interface will be provided to the users to rate the emails. When an email is opened that widget will appear inside the email. Users can provide their feedback from that. PHP scripting language is used for all the programming and MySQL for database programming. Following is the logic behind the email rating system to detect spam:

Every user initially has a score of 10. If a user press:

- "1 - not spam" = Score 3
- "2 - good" = Score 2
- "3 - may be" = Score 1
- "4 - bad" = Score 2
- "5 - spam" = Score 3

Nature of score assigned according to ratings
...-2, -1, 0, 1(spam), 2, 3, 4, 5(neutral), 6, 7, 8, 9, 10(ham), 11, 12...
Everything will be decided according to the final score calculated by given user ratings.

Example:

- User1 have 10 marks
- Press 5 = 10 - 3 = 7
- Press 2 = 7 + 2 = 9
- Press 3 = 9 + 1 = 10
- Press 5 = 10 - 3 = 7
- Press 4 = 7 - 2 = 5
- Press 1 = 5 + 3 = 8

After 6 ratings current mark is 8. Therefore, email is still ham (not spam).

In this system everything revolves arounds spammer or the sender's mail address. Because in existing systems email is the main element. but spam is still existing. So, I chose spammer as the main element of system.

14. Deliverables

14.1. SquirrelMail plugin

SquirrelMail is free and open source email client software. SquirrelMail supports plugins that can be used to extend the functionalities of the core system of SquirrelMail. In this project the concern was to develop such a plugin that can be used to get rating feedback from users.

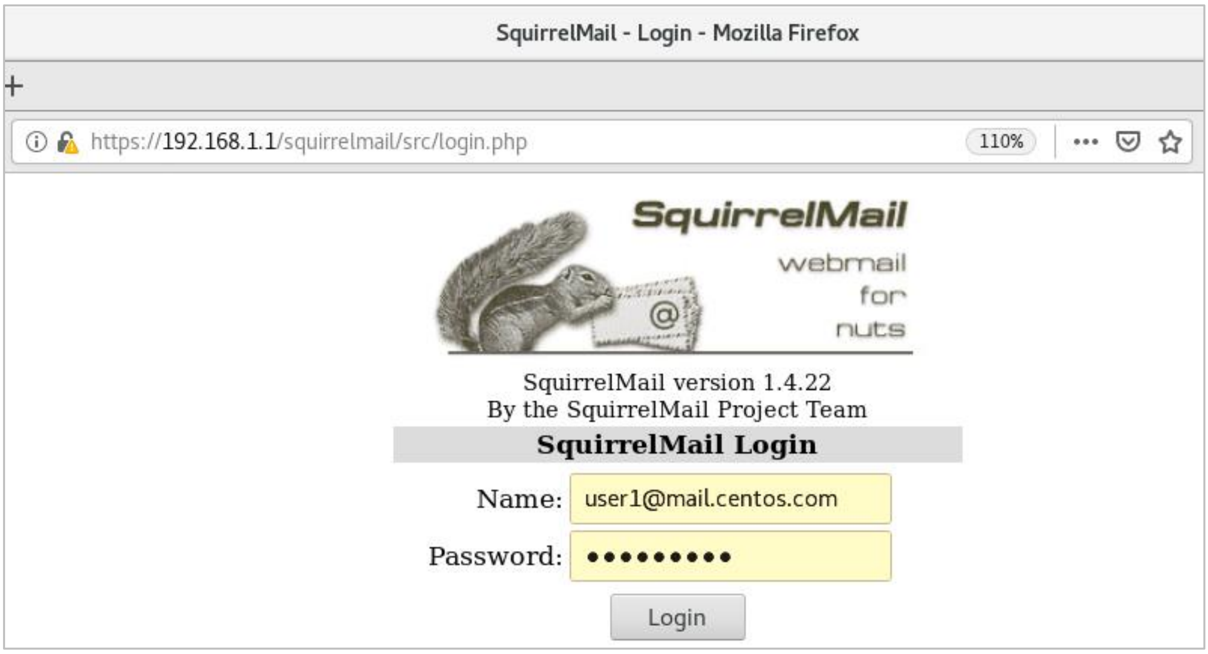
Basically, what is happening inside this plugin is assigning scores to email senders according to the ratings given by email users.

There are two types of scores:

1. “Spammer Mark”
2. “Mark”

“Mark” refers to the individual ratings given by different individual users according to their knowledge by looking at the email. If it passes a certain limit mail will be moved to junk folder. “Spammer Mark” refers to the score gained by sender of an email. It is used to decide whether email sender is a spammer or not. If it passes a certain limit sender will be consider as spammer and block.

14.1.1.Login page


SquirrelMail - Login - Mozilla Firefox
https://192.168.1.1/squirrelmail/src/login.php
110%
SquirrelMail
webmail
for
nuts

SquirrelMail version 1.4.22
By the SquirrelMail Project Team

SquirrelMail Login

Name: user1@mail.centos.com

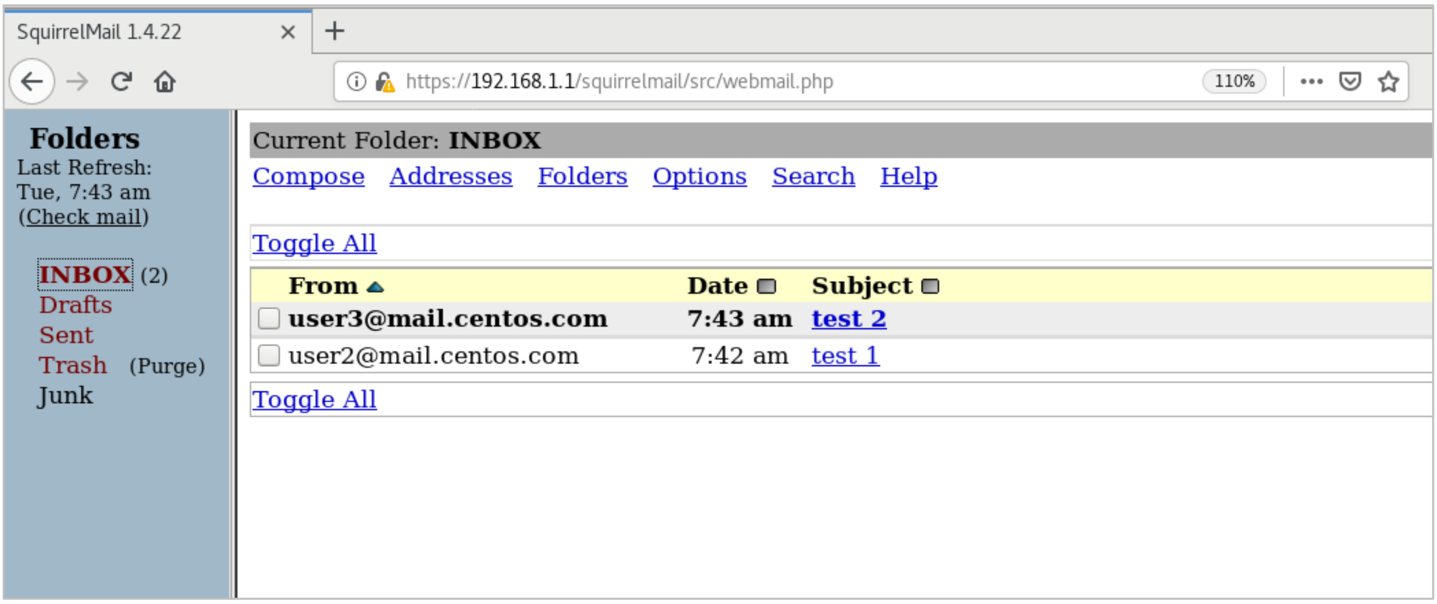
Password: ●●●●●●●●●

Login

Figure 22: SM login page

This is the login page of squirrel mail application. A user can provide his/her email and login with the account. This is running in local machine. There is also an email server running in localhost of this machine. Email server is iRedMail which is a combination of Postfix SMTP server and Dovecot IMAP server. For spam filtering and virus scanning SpamAssasin, ClamAV and Amavis were used.

14.1.2.Home page

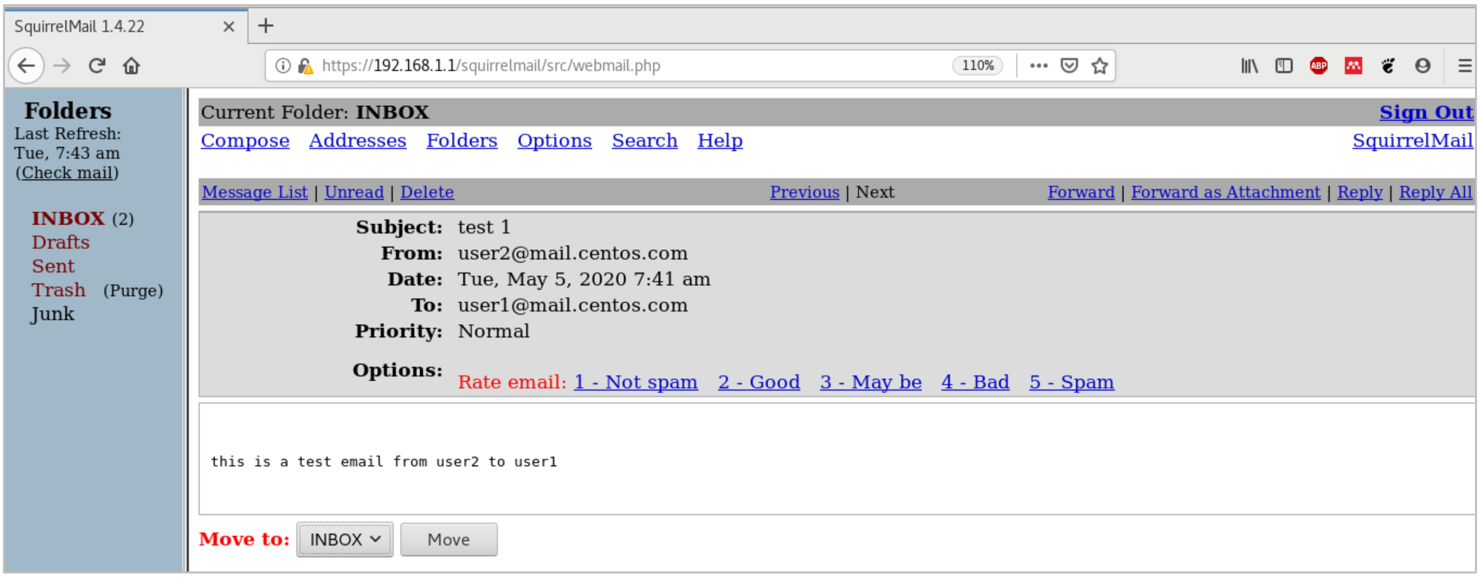


From	Date	Subject
□ user3@mail.centos.com	7:43 am	test 2
□ user2@mail.centos.com	7:42 am	test 1

Figure 23: SM inbox

This is the main page every user gets after login which is the inbox. In the above screenshot there are two emails received to user1's inbox, which were send by user2 and user3. Which shows that email server is working fine. Mail folders like inbox, drafts, sent, Trash and Junk are shown in the left side.

14.1.3.Received email



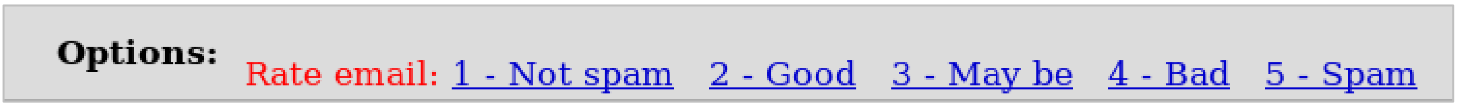
Subject: test 1
From: user2@mail.centos.com
Date: Tue, May 5, 2020 7:41 am
To: user1@mail.centos.com
Priority: Normal
Options: Rate email: 1 - Not spam 2 - Good 3 - May be 4 - Bad 5 - Spam

```
this is a test email from user2 to user1
```

Figure 24: SM inside of an email

This is the inner view of an email after opening. Subject, Sender, Receiver, Date, Options, Body are shown.

14.1.4.Email rating widget



Options: Rate email: 1 - Not spam 2 - Good 3 - May be 4 - Bad 5 - Spam

Figure 25: SM rating options

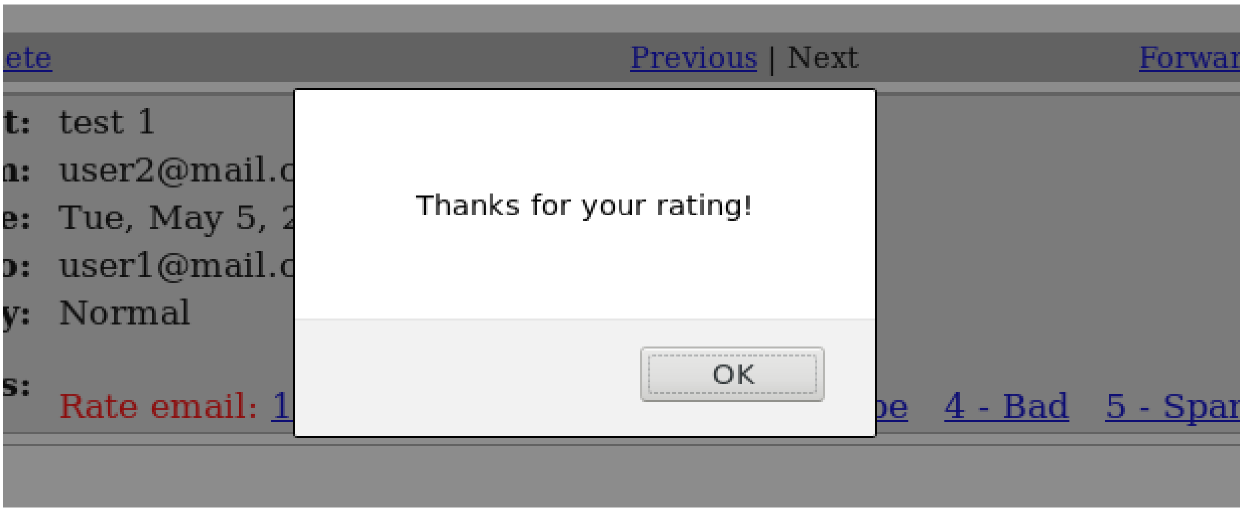
This is the option given to user to rate emails and provide feedback. There are five options with scores assigned to them:

1. Not spam -> score +3
2. Good -> score +2
3. May be -> score +/-1
4. Bad -> score -2
5. Spam -> score -3

A user can press any of the above given options in order to provide feedback.

- Every email sender initially receives a score of 10 which is previously called as “Mark”. If a user thinks this sender/email is not spam user can press **“1 – Not spam”**. Then 3 marks will be added to the sender’s score. If email is not a spam email a user can just ignore it but providing a feedback always help to optimize the spam filtering process. However, it’s always user’s choice.
- If a user thinks this sender/email contain somewhat spam materials and not completely genuine then user can press **“2 – Good”**. Then 2 marks will be added to the sender’s score.
- If a user can’t exactly decide the situation this sender/email, then user can press **“3 – May be”**. Then 1 will be added or subtracted from the sender’s score. When “Maybe” is pressed addition or subtraction depends on the current score of sender. If current score is less than or equal to five, 1 mark from five will be subtracted from current score. If current score is more than five, 1 mark will be added to the current score.
- If a user thinks this sender/email contain spam materials but not completely spam, then user can press **“4 – Bad”**. Then 2 marks will be subtracted from the sender’s score.
- If a user thinks this sender/email is completely spam and fake, then user can press **“5 – Spam”**. Then 3 marks will be subtracted from the sender’s score.

14.1.5.Prompt box



Thanks for your rating!

Figure 26: Prompt box

After pressing one of the above mentioned options. Prompt box in figure 5 will appear and after clicking “OK” user will be redirected to the Home page. Before redirecting ratings data will be sent to database.

14.1.6.Move option

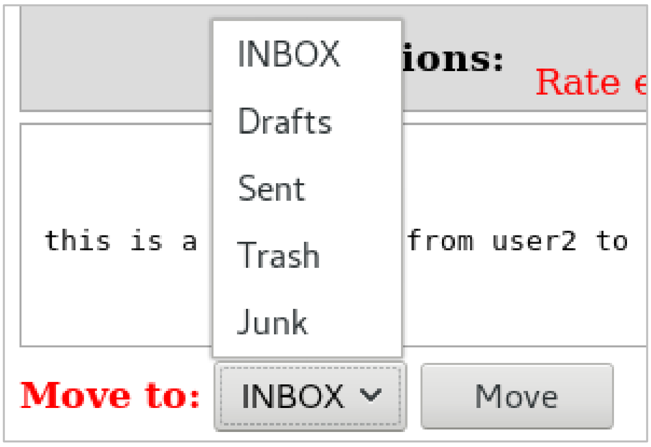
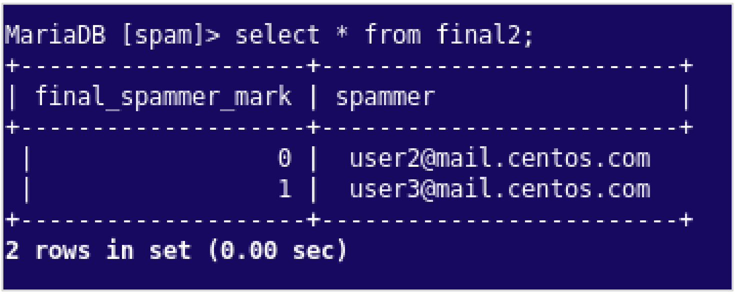


Figure 27: SM move option

If a user doesn’t want to see an email in Inbox or any other box, they can always move that email to another box relevant to that. This can be mainly used to move any spam mail to the Junk box immediately even without reporting as spam. This option is available at the end of every email body.

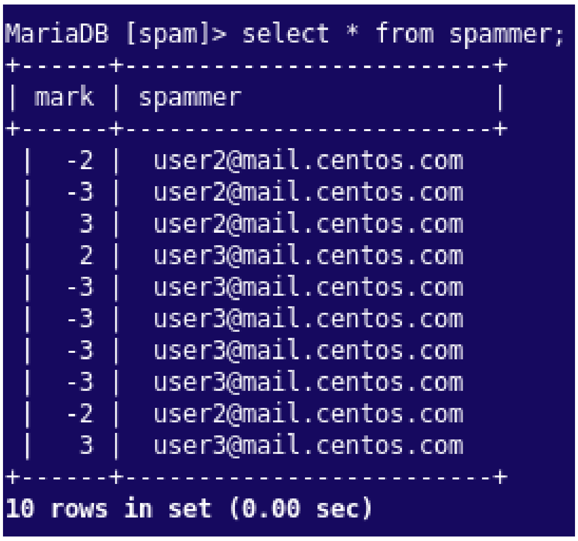
14.1.7.Database tables



```
MariaDB [spam]> select * from final2;
+-------------------+------------------------+
| final_spammer_mark | spammer                |
+-------------------+------------------------+
 |                 0 |  user2@mail.centos.com
 |                 1 |  user3@mail.centos.com
+-------------------+------------------------+
2 rows in set (0.00 sec)
```

Figure 28: SM db table final 2

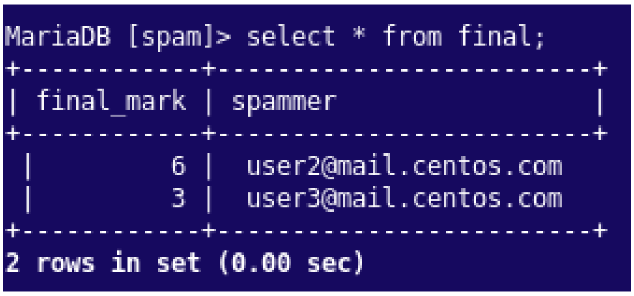
This table store “spammer mark” and sender(spammer) of an email. If this mark exceeds 3 that sender will be considered as a spammer and future email from that sender will be send to junk box.



```
MariaDB [spam]> select * from spammer;
+------+------------------------+
| mark | spammer                |
+------+------------------------+
 |   -2 |  user2@mail.centos.com
 |   -3 |  user2@mail.centos.com
 |    3 |  user2@mail.centos.com
 |    2 |  user3@mail.centos.com
 |   -3 |  user3@mail.centos.com
 |   -3 |  user3@mail.centos.com
 |   -3 |  user3@mail.centos.com
 |   -3 |  user3@mail.centos.com
 |   -2 |  user2@mail.centos.com
 |    3 |  user3@mail.centos.com
+------+------------------------+
10 rows in set (0.00 sec)
```

Figure 29: SM db table spammer

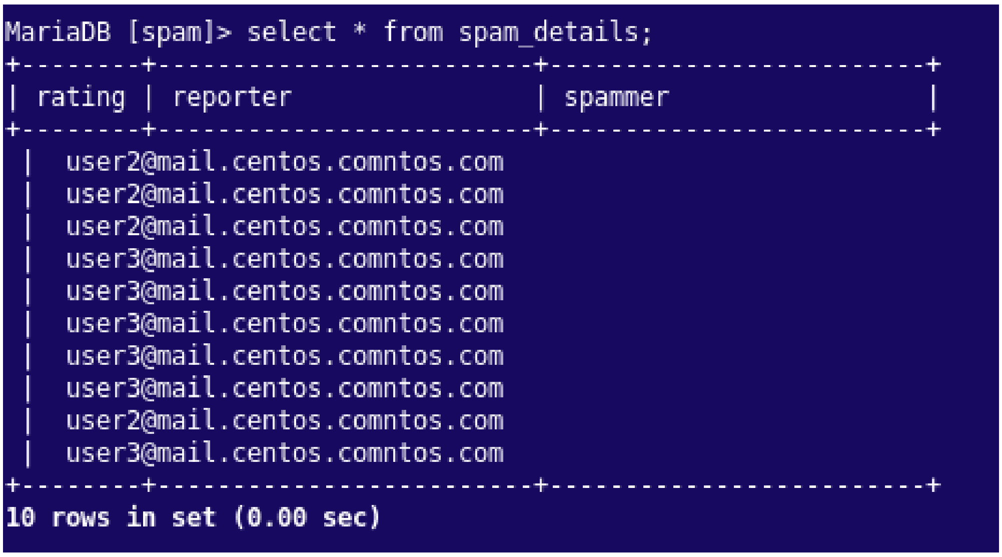
This table store “mark” and sender(spammer) of an email. These are the values given by individual users for the sender of email by clicking the five options in email rating widget. As you can see there is -3, -2 and 2 which means “spam”, “Bad” and “Good” respectively.



```
MariaDB [spam]> select * from final;
+------------+------------------------+
| final_mark | spammer                |
+------------+------------------------+
 |          6 |  user2@mail.centos.com
 |          3 |  user3@mail.centos.com
+------------+------------------------+
2 rows in set (0.00 sec)
```

Figure 30: SM db table final

This table store “Final mark” and sender(spammer) of an email. This value is calculated by “mark” which are ratings given by users. This “final mark” is used to decide a spammer.



```
MariaDB [spam]> select * from spam_details;
+--------+------------------------+------------------------+
| rating | reporter               | spammer                |
+--------+------------------------+------------------------+
 |  user2@mail.centos.comntos.com
 |  user2@mail.centos.comntos.com
 |  user2@mail.centos.comntos.com
 |  user3@mail.centos.comntos.com
 |  user3@mail.centos.comntos.com
 |  user3@mail.centos.comntos.com
 |  user3@mail.centos.comntos.com
 |  user3@mail.centos.comntos.com
 |  user2@mail.centos.comntos.com
 |  user3@mail.centos.comntos.com
+--------+------------------------+------------------------+
10 rows in set (0.00 sec)
```

Figure 31: SM db table spam details

This table stores “rating” which is “not spam”, “good”, “maybe”, “bad” and “spam” along with reporter of email and sender(spammer) of email. Output of the table isn’t correctly shown due to unknown error in MariaDB.

15. Project postmortem

15.1. Project overview

The objective of this project is to build a system consists of an extension to email client application and a spam filter. Extension can ask from user to give a rating to collect data about email to detect spam messages. Spam filter can use obtained email ratings to calculate what emails are being spam and not spam and block the spam messages before they arrive in majority of other email inboxes.

15.2. Key accomplishments

- As project design it worked well as intended. All the emails sent were received, ratings were processed and built functions works as required.
- Mainly built on the localhost to run locally. But ability access from a different operating system was successful.
- Database connectivity and network connectivity works fine.
- Ability to block and filter emails is successful.

15.3. Challenges faced

- When researching background information for this project, first I and my project supervisor came up with set of tools and software that can be used to setup and develop the system. They are Zimbra mail server and extension to Zimbra mail server which is called zimlet. After trying to get the setup to working state for a whole month I couldn't able to become successful. Then I switched to another set of tools and software with the advice and help from supervisor. A whole month got wasted.
- When starting to do this project I had little background information and knowledge about technologies that are going to use do this project. I spent at least 100 hours of background researching for related tools and technologies.
- I had very low familiarity with programming languages except C programming. This project required at least basic knowledge in different types of programming and scripting languages like MySQL, Python, Java, HTML, CSS, Javascript, Perl, PHP and Bash. I had to follow YouTube tutorials and read official documentations to understand some deep programming concepts from scratch.
- Time schedule was a huge pressure with other exams and coursework. I also happened to start the project little bit late.
- Working with open source software was very difficult. I had to read full documentations and source codes to understand the process and configurations of software applications.
- Working with virtual machines became very challenging due to lack of resources in my laptop to run multiple virtual machines at once.

15.4. Future considerations and improvements

- Use machine learning and artificial intelligence to improve efficiency and performance.
- Artificial intelligence can be used to scan real time conversations on public forums and news articles to understand any ongoing or developing trend in spam messages around the world.
- Integration of this community based methodology with other existing techniques to see much more accurate and efficient outcome.

15.5. Lessons learned

- How to write a research paper, project report, project proposal and other project related official documents.
- Importance of spam filtering in the real world.
- New programming languages, concepts and more advanced programming methodologies. (ex: Python, Perl, PHP, Bash)
- Understood that programming is much more related and essential for networking and cybersecurity than I thought.
- More advanced concepts and underlying processes about Linux operating system and got good knowledge with hands on experience with Linux.
- How email works and how different protocols like SMTP, POP, IMAP works in background to deliver email to our inbox.
- Linux troubleshooting, and how to work with open source software and GitHub.
- Importance of having a well-managed and planned schedule to complete a project.
- Importance of having a well-defined scope and its importance to complete the project within given schedule and resources.

16. Conclusion

As a conclusion, the overall functionality of community based spam filtering system works fine. Integrated components like SMTP server. IMAP server and email client application works together smoothly. Intended task of the project has been achieved.

17. References

• *Number of e-mail users worldwide 2023 | Statista* (no date). Available at: https://www.statista.com/statistics/255080/number-of-e-mail-users-worldwide/ (Accessed: 22 February 2020).

Apache SpamAssassin: Welcome (no date). Available at: https://spamassassin.apache.org/ (Accessed: 5 March 2020).

Banday, M. T. and Jan, T. R. (2009) ‘Effectiveness and Limitations of Statistical Spam Filters’, pp. 1–13. Available at: http://arxiv.org/abs/0910.2540.

Best Open Source E-Mail-Server (no date). Available at: https://www.psychz.net/client/blog/de/best-open-source-email-server.html (Accessed: 24 February 2020).

Bhowmick, A. and Hazarika, S. M. (2016) ‘Machine Learning for E-mail Spam Filtering: Review,Techniques and Trends’, (January). doi: 10.1007/978-981-10-4765-7.

Blanzieri, E. and Bryl, A. (2008) ‘A survey of learning-based techniques of email spam filtering’, p. 30.

Can artificial intelligence spot spam quicker than humans? (no date). Available at: https://www.information-age.com/artificial-intelligence-spam-machine-learning-123481368/ (Accessed: 23 February 2020).

Christina, V., Karpagavalli, S. and Suganya, G. (2010) ‘A Study on Email Spam Filtering Techniques’, *International Journal of Computer Applications*, 12(1), pp. 7–9. doi: 10.5120/1645-2213.

Cyber Attacks Cost $45 billion in 2018 | 2019-07-10 | Security Magazine (no date). Available at: https://www.securitymagazine.com/articles/90493-cyber-attacks-cost-45-billion-in-2018 (Accessed: 21 February 2020).

Daily number of e-mails worldwide 2023 | Statista (no date). Available at: https://www.statista.com/statistics/456500/daily-number-of-e-mails-worldwide/ (Accessed: 23 January 2020).

Defense Minister Was on the Line, Asking for Millions to Aid France. Or Was He? - The New York Times (no date). Available at: https://www.nytimes.com/2020/02/04/world/europe/france-Jean-Yves-Le-Drian-fraud.html (Accessed: 21 February 2020).

Documentation - MailCleaner - Open source Anti spam & Antivirus gateway (no date). Available at: https://www.mailcleaner.org/documentation/ (Accessed: 24 February 2020).

E-mail spam rate worldwide 2018 | Statista (no date). Available at: https://www.statista.com/statistics/270899/global-e-mail-spam-rate/ (Accessed: 5 February 2020).

efa_project_v4 [eFa project Wiki] (no date). Available at: https://wiki.efa-project.org/doku.php?id=efa_project_v4 (Accessed: 24 February 2020).

eFa project – email FILTER appliance project (no date). Available at: https://efa-project.org/ (Accessed: 5 March 2020).

Eric Allman (no date) *Sendmail - Installation Guide*. Available at: https://www.sendmail.org/~ca/email/doc8.9/op.html (Accessed: 24 February 2020).

Field, J. (no date) *MailScanner User Guide and Training Manual*.

Future trends in spam filtering | datascienceCMU (no date). Available at: https://datasciencecmu.wordpress.com/2014/04/18/future-trends-in-spam-filtering/ (Accessed: 23 February 2020).

Global email platform market share 2018 | Statista (no date). Available at: https://www.statista.com/statistics/709596/most-used-e-mail-platform-by-market-share/ (Accessed: 24 January 2020).

Home - MailScanner (no date). Available at: https://www.mailscanner.info/ (Accessed: 5 March 2020).

Horrigan, J. B., Rainie, L. and S, F. (2001) *Online communities: Networks that nurture long-distance relationships and local ties*. Available at: http://www.pewinternet.org/pdfs/Report1.pdf.

How Email Really Works (no date). Available at: https://www.oasis-open.org/khelp/kmlm/user_help/html/how_email_works.html (Accessed: 2 February 2020).

Internet Crime Complaint Center (IC3) | Business E-mail Compromise The 12 Billion Dollar Scam (no date). Available at: https://www.ic3.gov/media/2018/180712.aspx (Accessed: 5 February 2020).

Internet Crime Complaint Center (IC3) | IC3 2011 Annual Report on Internet Crime Released (no date). Available at: https://www.ic3.gov/media/2012/120511.aspx (Accessed: 24 January 2020).

iredmail logo - Google Search (no date). Available at: https://www.google.com/search?q=iredmail+logo&client=firefox-b-d&sxsrf=ALeKk03ZG9GQPRjlV6Zi_RNVX0Fl00D0RQ:1583429367754&tbm=isch&source=iu&ictx=1&fir=Ji6iRPfBGVlr8M%253A%252C7cN896S1GIIsAM%252C_&vet=1&usg=AI4_-kQaizVuONTpjZb9Wp2oj0QOuY8UAw&sa=X&ved=2ahUKEwiipPeH7oPoAhXEfn0KHTkvDfoQ9QEwAXoECAoQHQ#imgrc=Ji6iRPfBGVlr8M: (Accessed: 5 March 2020).

Mail (MX) Server Survey (no date). Available at: http://www.securityspace.com/s_survey/data/man.201801/mxsurvey.html (Accessed: 21 February 2020).

mail cleaner logo - Google Search (no date). Available at: https://www.google.com/search?q=mail+cleaner+logo&client=firefox-b-d&sxsrf=ALeKk01OzRUXTZvf-aM6qNeImHtQ4H6cig:1583428328677&tbm=isch&source=iu&ictx=1&fir=ZPj2NVo862UilM%253A%252Cb6HNUOqwV0QeBM%252C_&vet=1&usg=AI4_-kR9SGhk92BKueEHGBBtWctMaZZ7MQ&sa=X&ved=2ahUKEwizgruY6oPoAhUT63MBHWnaBH8Q9QEwAHoECAoQEA#imgrc=ZPj2NVo862UilM: (Accessed: 5 March 2020).

Nazirova, S. (2011) ‘Survey on Spam Filtering Techniques’, *Communications and Network*. doi: 10.4236/cn.2011.33019.

Number of Gmail active users 2018 | Statista (no date). Available at: https://www.statista.com/statistics/432390/active-gmail-users/ (Accessed: 24 January 2020).

OrangeAssassin | New Open Source Antispam Email Filtering Framework (no date). Available at: https://orangeassassin.org/ (Accessed: 24 February 2020).

orangeassassin logo - Google Search (no date). Available at: https://www.google.com/search?q=orangeassassin+logo&client=firefox-b-d&sxsrf=ALeKk00rsOrwRAaWM_GQ9KYfGxODRcOC-A:1583429287794&tbm=isch&source=iu&ictx=1&fir=iPhy5U8ONdOBwM%253A%252COlQb6fgUtuFuSM%252C_&vet=1&usg=AI4_-kQyfJY8WBcYad_kCU_cZkQXbu768Q&sa=X&ved=2ahUKEwj8zubh7YPoAhXDZSsKHeSODRwQ9QEwA3oECAoQCw#imgrc=iPhy5U8ONdOBwM: (Accessed: 5 March 2020).

Phishing Line - IT Strategy & Architecture (no date). Available at: http://blogs.plymouth.ac.uk/strategyandarchitecture/phishing-line/ (Accessed: 21 February 2020).

Postfix (software) - Wikipedia (no date). Available at: https://en.wikipedia.org/wiki/Postfix_(software) (Accessed: 24 February 2020).

postfix logo - Google Search (no date). Available at: https://www.google.com/search?q=postfix+logo&client=firefox-b-d&sxsrf=ALeKk00droxCQ3nED27JoIJLRegcHUBGCw:1583429454703&tbm=isch&source=iu&ictx=1&fir=2qWKhfixtLUdJM%253A%252CKKBOs_MC1bxBrM%252C_&vet=1&usg=AI4_-kS2NOM2d_btZAK03y0cn5jU6tSKjQ&sa=X&ved=2ahUKEwjeqLKx7oPoAhWUUn0KHd9pDjAQ9QEwAHoECAoQFA#imgrc=2qWKhfixtLUdJM: (Accessed: 5 March 2020).

sendmail logo - Google Search (no date). Available at: https://www.google.com/search?q=sendmail+logo&tbm=isch&ved=2ahUKEwjdks2Y7oPoAhUO8TgGHeItBZoQ2-cCegQIABAA&oq=sendmail+logo&gs_l=img.3..0.27611.29624..29904...0.0..0.220.1453.0j2j5......0....1..gws-wiz-img.......0i67j0i7i30.p-_NJ8qZSO0&ei=GjdhXt2dK47i4-EP4tuU0Ak&client=firefox-b-d#imgrc=qdrIg6QLvXB5IM (Accessed: 5 March 2020).

Spam e-mail: countries of origin 2019 | Statista (no date). Available at: https://www.statista.com/statistics/263086/countries-of-origin-of-spam/ (Accessed: 5 February 2020).

Spam statistics: spam e-mail traffic share 2019 | Statista (no date). Available at: https://www.statista.com/statistics/420391/spam-email-traffic-share/ (Accessed: 23 January 2020).

SpamAssassin: Documentation (no date). Available at: https://spamassassin.apache.org/doc.html (Accessed: 24 February 2020).

SquirrelMail - Webmail for Nuts! (no date). Available at: http://squirrelmail.org/docs/admin/admin-1.html (Accessed: 24 February 2020).

squirrelmail logo - Google Search (no date). Available at: https://www.google.com/search?q=squirrelmail+logo&client=firefox-b-d&sxsrf=ALeKk02M267GlJOnkCGc99lAYStybTUl0w:1583429400068&tbm=isch&source=iu&ictx=1&fir=HrHU3nYDwcV8qM%253A%252CxGMCGd6I8wjq-M%252C_&vet=1&usg=AI4_-kRvNXe6JImtqguAzdg21tjJ_MNl7Q&sa=X&ved=2ahUKEwjpx6uX7oPoAhWVlEsFHQWvCvYQ9QEwAHoECAkQFQ#imgrc=HrHU3nYDwcV8qM: (Accessed: 5 March 2020).

The Top Spam Filtering Companies - February 2020 | 99firms (no date). Available at: https://99firms.com/spam-filtering-companies/#gref (Accessed: 22 February 2020).

The World’s Leading Email Collaboration Platform - Zimbra (no date). Available at: https://www.zimbra.com/ (Accessed: 5 March 2020).

What is email spam? - Definition from WhatIs.com (no date). Available at: https://searchsecurity.techtarget.com/definition/spam (Accessed: 2 February 2020).

18. Bibliography

Secure and Open Unified Collaboration Platform. 2020. Zimbra Documentation. [ONLINE] Available at: https://www.zimbra.com/documentation/. [Accessed 23 January 2020].

Ubuntu Desktop Guide. 2020. Ubuntu Desktop Guide. [ONLINE] Available at: https://help.ubuntu.com/lts/ubuntu-help/index.html. [Accessed 23 January 2020].

Introduction to Linux. 2020. Introduction to Linux. [ONLINE] Available at: https://linux.die.net/Intro-Linux/. [Accessed 23 January 2020].

Oracle Help Center. 2020. Java Documentation - Get Started. [ONLINE] Available at: https://docs.oracle.com/en/java/. [Accessed 23 January 2020].

JavaScript | MDN (no date). Available at: https://developer.mozilla.org/en-US/docs/Web/JavaScript (Accessed: 17 February 2020).

3.8.2rc2 Documentation (no date). Available at: https://docs.python.org/3/ (Accessed: 21 February 2020).

PHP: PHP Manual - Manual (no date). Available at: https://www.php.net/manual/en/ (Accessed: 21 February 2020).

iRedMail Documentations (no date). Available at: https://docs.iredmail.org/ (Accessed: 21 February 2020).

Introduction — OrangeAssassin 1.0a1 documentation (no date). Available at: https://orangeassassin.readthedocs.io/en/v1.0b/intro.html (Accessed: 21 February 2020).

CentOS Documentation Home :: CentOS Docs Site (no date). Available at: https://docs.centos.org/en-US/docs/ (Accessed: 21 February 2020).

Postfix Documentation (no date). Available at: http://www.postfix.org/documentation.html (Accessed: 21 February 2020).

Anti-spam: 8 benefits for corporate use - OSTEC Blog (no date). Available at: https://ostec.blog/en/general/anti-spam-8-benefits-for-corporate-use (Accessed: 22 February 2020).

Ten Spam-Filtering Methods Explained | TechSoup Canada (no date). Available at: https://www.techsoupcanada.ca/en/learning_center/10_sfm_explained (Accessed: 25 January 2020).

3rd, D. E. E. and Manning, B. (no date) ‘Domain Name System (DNS) IANA Considerations’.

RFC 2929 - Domain Name System (DNS) IANA Considerations (no date). Available at: https://tools.ietf.org/html/rfc2929 (Accessed: 22 February 2020).

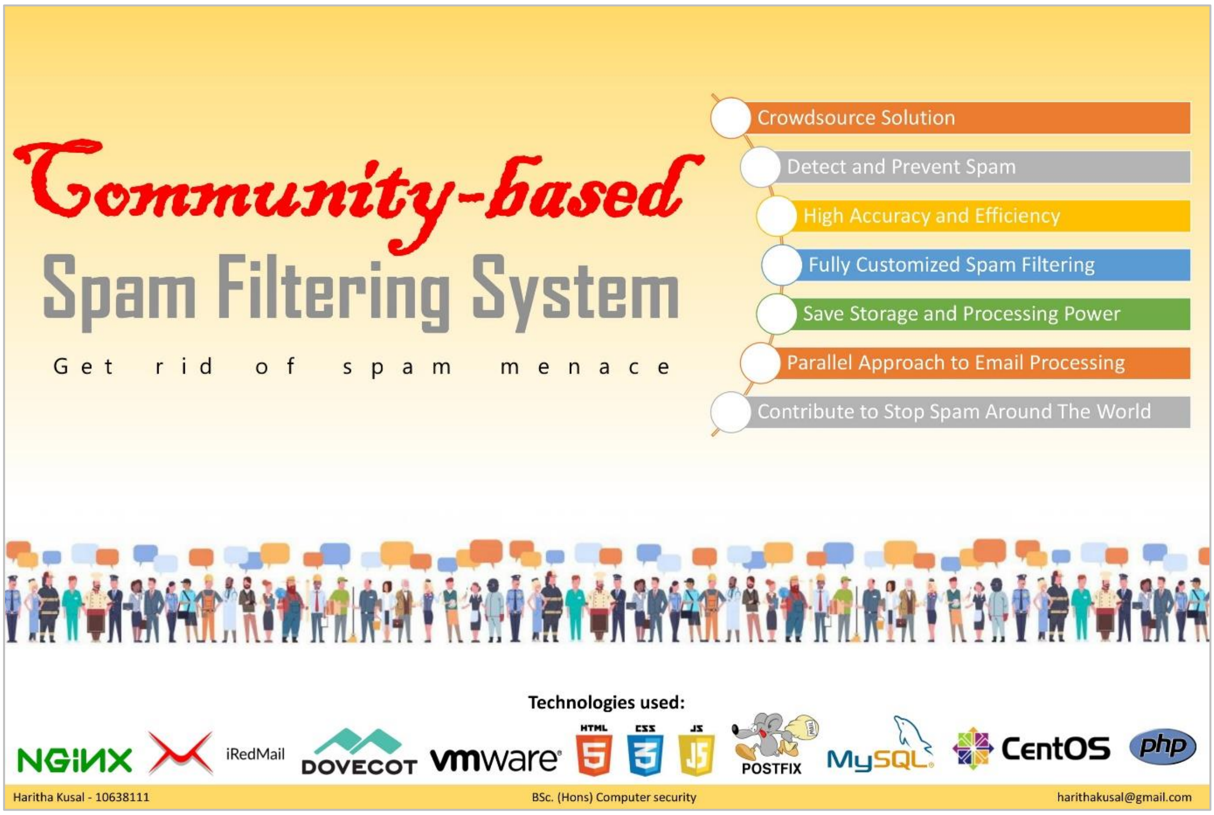
RFC 5321 - Simple Mail Transfer Protocol (no date). Available at: https://tools.ietf.org/html/rfc5321 (Accessed: 22 February 2020).

RFC 2131 - Dynamic Host Configuration Protocol (no date). Available at: https://tools.ietf.org/html/rfc2131 (Accessed: 22 February 2020).

RFC 2076 - Common Internet Message Headers (no date). Available at: https://tools.ietf.org/html/rfc2076 (Accessed: 22 February 2020).

19. Appendices

19.1. Poster



Community-based

Spam Filtering System

Get rid of spam menace

- Crowdsource Solution
- Detect and Prevent Spam
- High Accuracy and Efficiency
- Fully Customized Spam Filtering
- Save Storage and Processing Power
- Parallel Approach to Email Processing
- Contribute to Stop Spam Around The World

Technologies used:

Haritha Kusal - 10638111

BSc. (Hons) Computer security

harithakusal@gmail.com

Figure 32: poster

19.2. System diagrams

19.2.1. Use case diagram

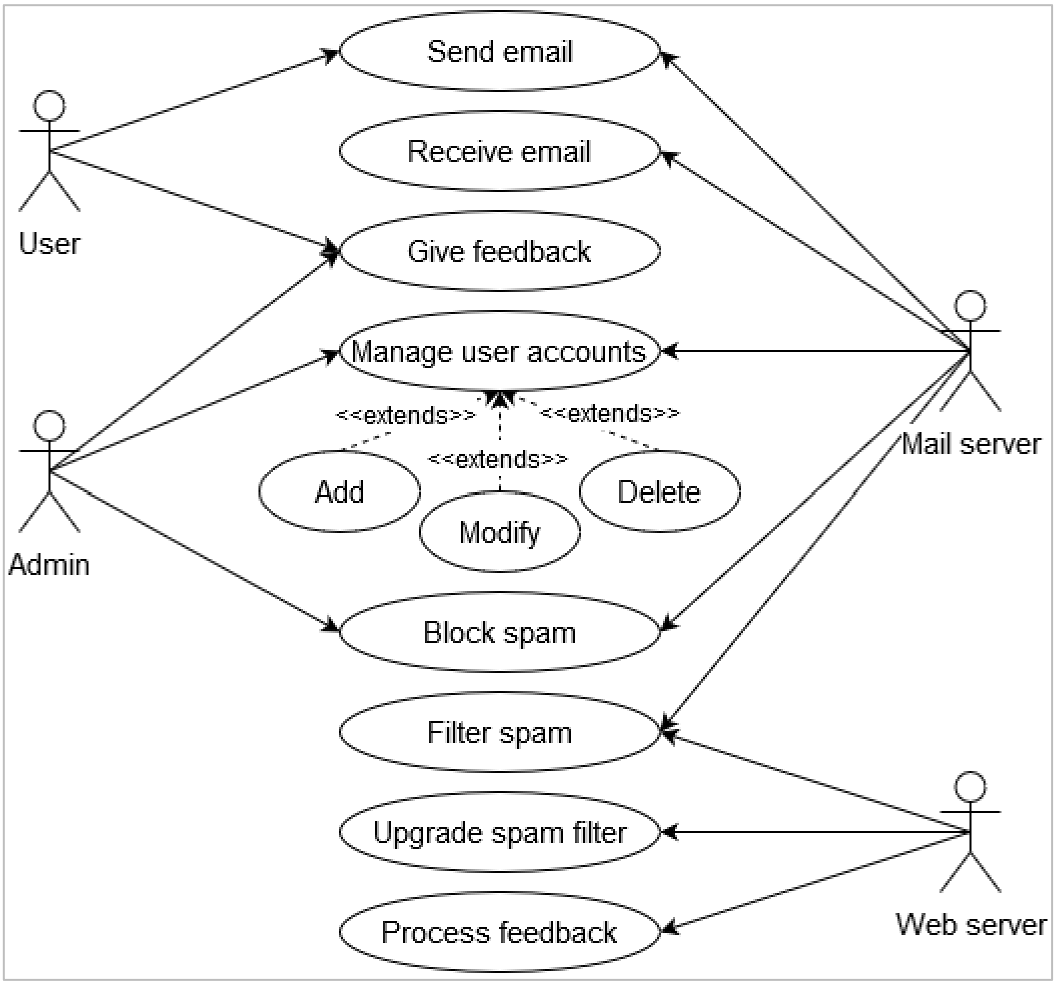


Figure 33: Use case diagram

19.2.2.Activity diagram

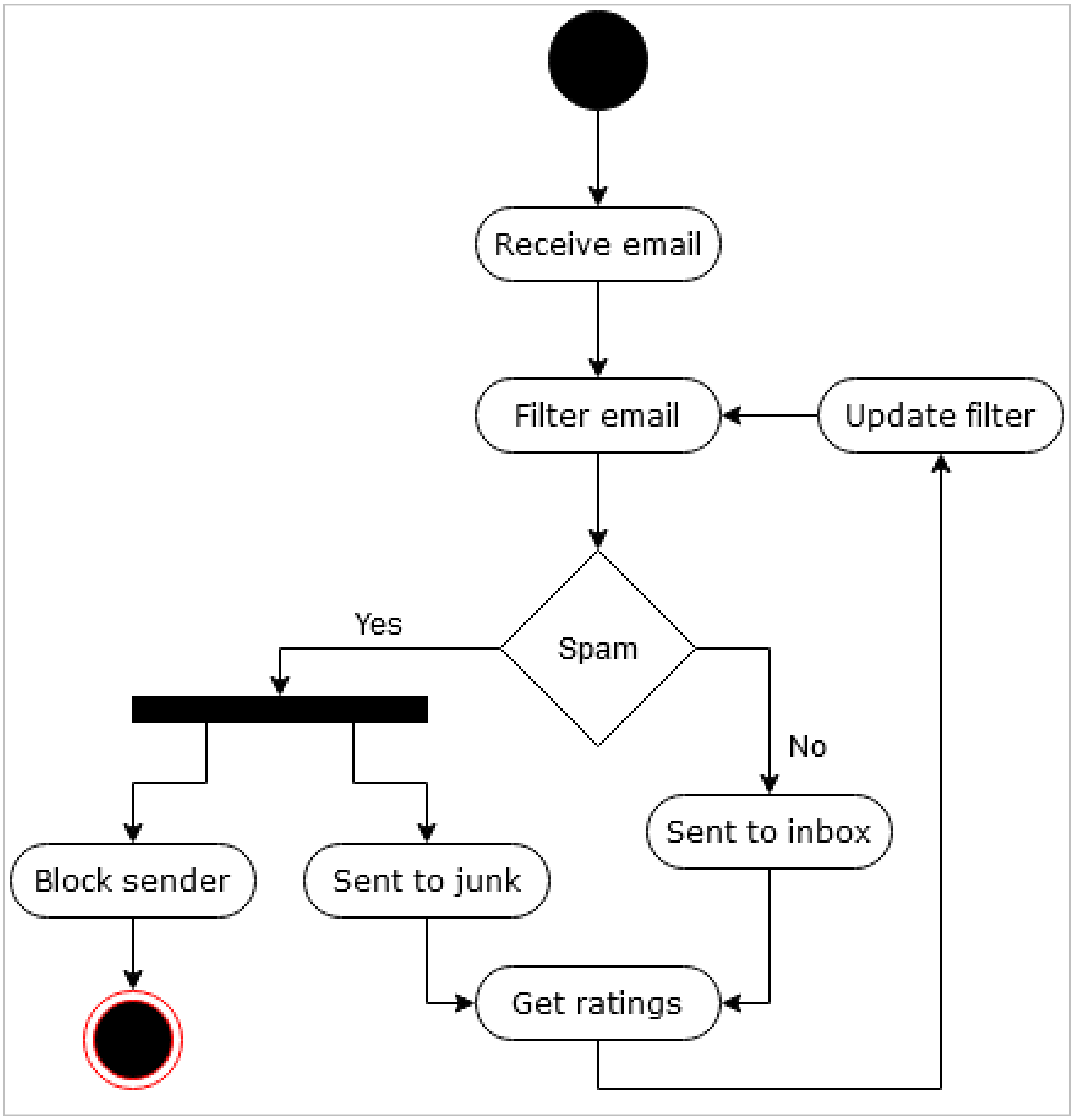


Figure 34: Activity diagram

19.3. PID

1. Introduction

For the purpose of this module PRCO303SL, my target is to create a client-side spam filtering application. The objective of this effort is to make a spam filtering application with a much lower false-positive rate. The outcome ought to be a client-side spam filter, that is easy to operate by user and robust in work. The principal requirements were first recognized to develop a good final product. In the first place, the program should bring about less inadvertently blocking genuine messages as its cost is a lot higher than giving spam a chance to bypass the security. The below needs express that the program should concentrate on inbound emails and utilize diverse spam and ham detection techniques. Utilizing various systems one after another builds the precision of recognizing spam messages. This client-side spam filter utilizes different separate strategies. Initial one dependent on *Naive Bayesian classifier* that takes a shot at the watchword context. The watchword context was made by mapping a current watchword to a set of associated words. The subsequent filter dependent on *grey listing, blacklisting and whitelisting*. The motivation behind the combination of such methods was to give precise outcomes to sort the message to be spam or non-spam.

2. Motivation

These days individuals live in a world of innovation and information technology, where computers are a significant piece of our lives, and the Internet is essential for work, study or just for relaxation. Email service is a productive and the least expensive method for communication. The quick improvement of information and communication innovation brings more points of interest, yet additionally certain issues, for instance, spam. The upsurge in the volume of unwanted messages called spam has made a serious requirement for the improvement of progressively trustworthy and vigorous antispam programs. Spammers constantly develop their techniques to arrive at more inboxes by bypassing the email filters. Therefore, it is imperative to utilize powerful spam filters that can remain ahead of the spammers and keep on giving organizations security from email malware and malicious programs.

One of my major concerns in developing this application is to research and self-learn concepts and technologies behind this project to improve my knowledge and experience regarding computer security. My effort to create a unique anti-spam software program for a final year project has been improved due to information gathered from my own research.

I am motivated to find a method that enhances spam filtering in a way that,

- Reduce the number of grey mails.
- Increase the accuracy of spam detection.
- Lower false-positive rate.
- Provide a simple and robust client-side spam filtering application.

- Provide user-friendly and configurable user interfaces.
- Improve the efficiency of spam filtering.

Even though I don't have an exclusive real-world client who has business needs that my product is targeting to fulfill, the World is full of potential clients and users with a multibillion-dollar industry that is hungry for this kind of application.

3. Project Objectives

- Analyze existing spam filtering methods and provide recommendations for improvement.
- Analyze requirements for the new spam filtering application in line with the new methodologies and technologies.
- Analyze possible development technologies and deployment solutions
- Implement a new spam filtering application in line with new methodologies and technologies.

4. Initial Scope

- The scope and objectives were identified by thorough background research.
- The program is planned to check one message at a time and deliver results.
- Develop a graphical user interface using programming language to get user data.
- Provide different options for the user to choose from which method is preferring to filter messages.
- Users can rate messages and check whether the message is spam or not.
- Spam filter run the content of the provided message through given message and identify message as spam or non-spam.

5. Method of Approach

The project will be completed in different stages,

i. Identifying project scope and background research.
ii. Develop back-end algorithms.
iii. Design and develop user interfaces.
iv. Train and test final program

Possible technologies – MySQL, Java, NetBeans IDE, HTML, CSS, Javascript, PHP

6. Initial Project Plan

Project plan		
Stage	**Deadline**	**Outcome**
1. Background research	09/10/2019	Project proposal
2. Initiation	03/11/2019	Project initiation document
3. High-level design	12/11/2019	Design plan (DB schema, GUI style, Pseudo codes, Architecture)
4. Increment 1	18/12/2019	Back-end (algorithms)
5. Increment 2	13/01/2020	Front-end (user interfaces)
6. Increment 3	17/02/2020	The integrated backend and front-end
7. Testing and evaluation	20/03/2020	Assembled product and statistical evaluation
8. Finalizing	01/04/2020	Final product and report

7. Control Plan

The following control techniques will be employed,

- End-Stage reports
- End-Stage review with supervisor
- Interim reports
- Risk management
- Communication plan
- Quality plan

8. Communication Plan

In addition to ad-hoc supervisor meetings as necessary, planned review/feedback meetings will be held at the end of each stage in order to discuss the end-Stage report, the Next Stage plan, and to review any technical deliverables produced during the stage. Feedback meetings will also be held following the submission of the two Interim reports.

9. Initial Risk List

Risk	**Management strategy**
Schedule overrun	Having a contingency plan
Technology failure	Take backups and use standard technologies
Difficulty in using new technologies	Develop a simple prototype first
Lack of project management skills	Seek to advise from supervisor
Unclear/misunderstood scope	Continuous evaluation and background research

10. Initial Quality Plan

Quality check	**Strategy**
Scope validation	Background research, Continuous evaluation of the project in all stages
Design validation	Check against scope, DB and software principles at the ned of increment 2
Usability	To be conducted at the end of each incremental stage
Testing	To be conducted in stage 7 and onwards

19.4. Final project plan

Stage	Deadline	Outcome
Background research	30/10/2019	Research of background knowledge to understand project work.
Project initiation	05/11/2019	Making project initiation document and defining cope and boundaries.
High level design	28/12/2019	Create high level design and plan (ex: UML diagrams)
Setting up the environment	27/01/2020	Setup the system needed for development and demonstration of the project
Development of spam engine	25/03/2020	Create spam engine
Development of email rating system	20/04/2020	Create email rating system
Testing and evaluation	30/04/2020	Test, correct errors and evaluate the outcome of the project

19.5. Interim report 1

1. Purpose

This report details the processes of the system going to design, build, and test a Spam filtering application. The objective of this effort is to make a spam filtering application with a much lower false-positive rate. The outcome ought to be a client-side spam filter, that is easy to operate by user and robust in work. The principal requirements were first recognized to develop a good final product. In the first place, the program should bring about less inadvertently blocking real ham messages as its cost is a lot higher than giving spam a chance to bypass the security. I want to make our email inbox less annoying and more helpful.

2. Background

Nowadays, communication via email is very prominent. Everyone has an email ID, and everyone sends and receives emails. Spammers use bulk email technologies to send thousands of emails at once to random users. Roughly 75% of global email traffic is now spam, which means for one legitimate email you receive four spam emails. Such a huge amount of spam makes servers busy by processing spam and waste valuable resources. Organizations with 24 users or less receive 600 spam messages per month. Those unnecessary traffic cause many problems like reducing bandwidth, lowering service performances, and reducing the productivity of employees.

3. Motivation

I have a setup of four email IDs on my computer and mobile phone. Two of which are my personal emails and another two from NSBM University and Plymouth University. Every day when I investigate the email box, I see at least 3 spam emails. Even though I know those are spam messages still I want to look at what are they and who had sent them, this creates distractions every time when I am going to check my emails. So, I thought about creating a spam filtering application for my final project. I may not be able to create a commercial level spam filter, but at least I want to give it a try to create a new technique to stop email spams. In the project, my idea is to Implement a demo system for a spam filtering application.

4. Goal

The goal of this project is to provide a simple, easy to use and customizable software application to enhance the spam filtering process with a new technique. This application will not be a replacement for your typical regular email application. My intent is to *create a security technique* for spam filtering not to create a spam filtering software.

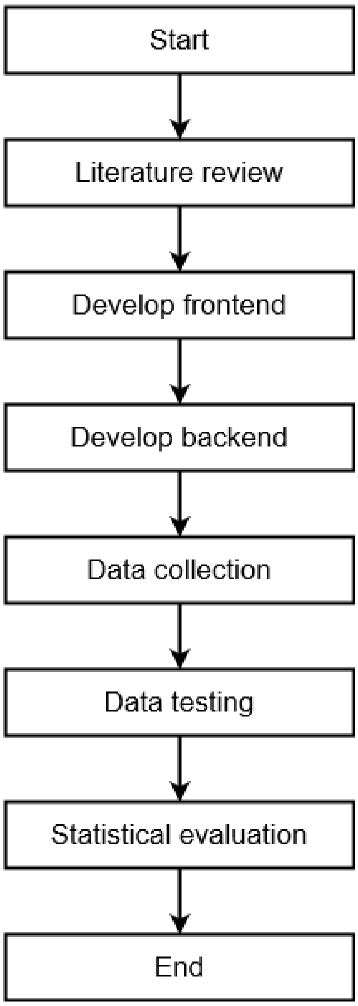
5. System description

The first user should install the application in his/her device, in this case, a desktop computer. Because I am developing this application for windows desktop only. After finishing the installation user can open the application and upload emails that are required to test for spam. Then the user can select which algorithm they want to use for spam filtering purposes.

They can use both naïve Bayesian technique or listing algorithm at once or one algorithm at a time. After selecting the algorithm results can be obtained after the filtration process. The results will be displayed below separately for both algorithms.

6. Project methodology

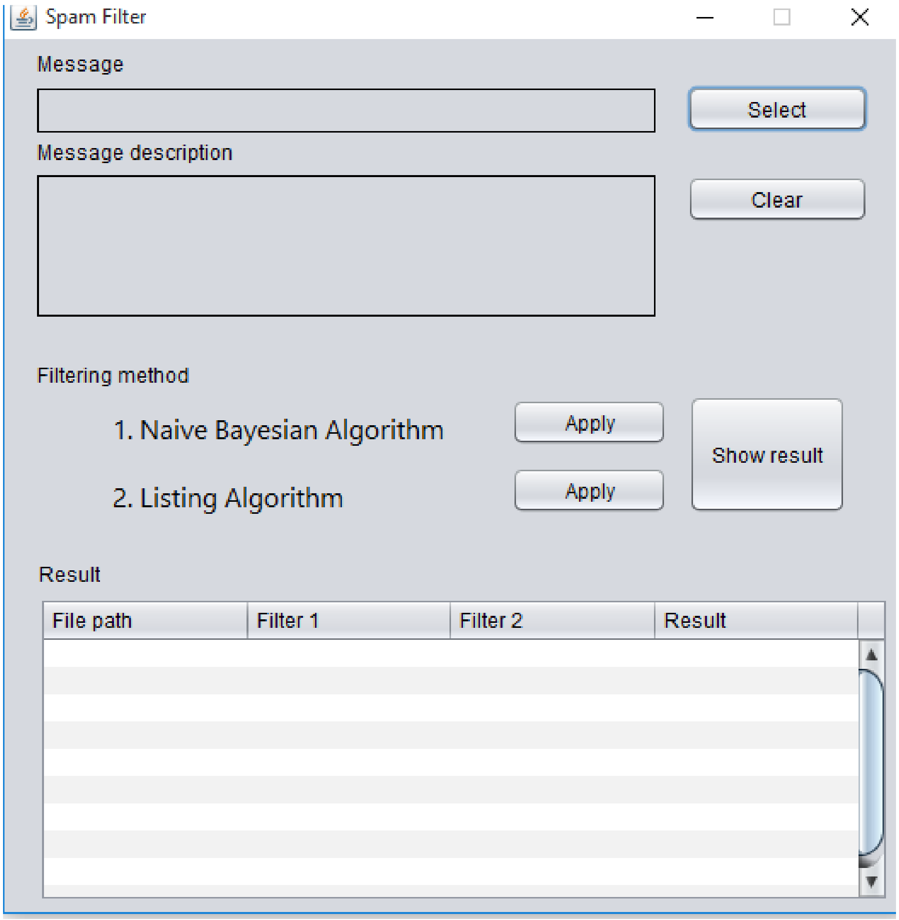
Spam filtering application consists of the main two important parts which were the frontend development and backend development. This whole development process involves six stages. Starting from the literature review which allows me to do background research and acquire knowledge about the project I'm going to do. Next, developing the frontend, which is the graphical user interface, which is the easiest phase, then developing backend which is the toughest and most advanced phase, next is data collection which allows collecting data to test, then data testing and statistical evaluation.



7. System development

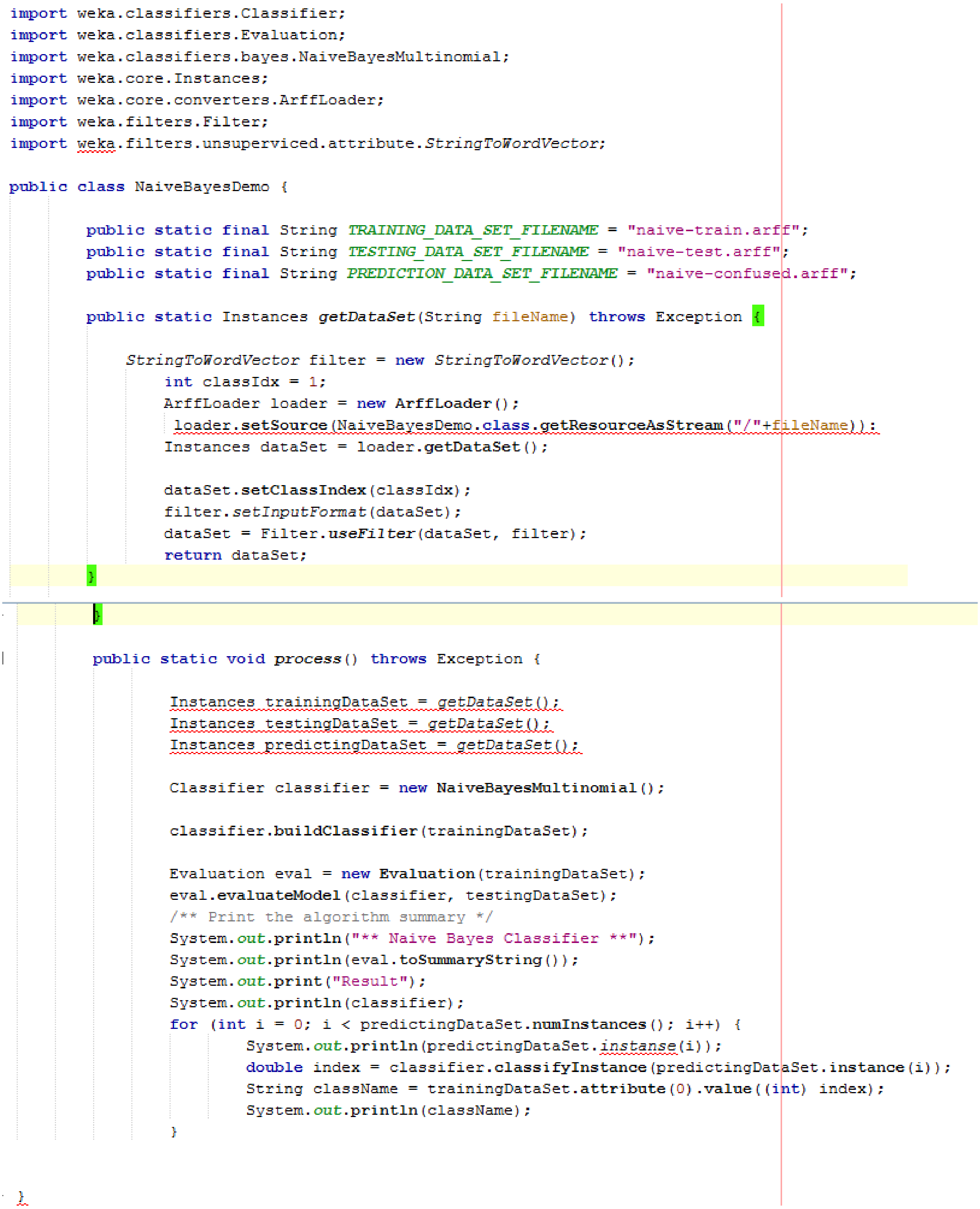
7.1 User interface development

I am using Oracle Netbeans IDE for development purposes with Java programming language. The main user interface has been already created. You can see below a screenshot of the user interface. “Select” button will allow you to select and upload emails that are needed to be tested to the application. “Clear” button will clear all the uploaded content. “Apply” buttons will select algorithms simultaneously or both at once. “Show result” button will show the intended result after calculation.



7.2 Naïve Bayesian algorithm development

First, I solved the Bayesian algorithm with an example, then wrote pseudocode. Then I started to program the algorithm. I haven’t completed the algorithm yet. Naïve Bayesian algorithm will check the possibility of an email being spam or ham using a mathematical model. Naïve Bayes classifiers are a group of calculations dependent on Bayes' Theorem. It's not only one algorithm but a collection of different algorithms with operating under a common rule, for example, each pair of data being calculated is autonomous of one another.



```java
import weka.classifiers.Classifier;
import weka.classifiers.Evaluation;
import weka.classifiers.bayes.NaiveBayesMultinomial;
import weka.core.Instances;
import weka.core.converters.ArffLoader;
import weka.filters.Filter;
import weka.filters.unsuperviced.attribute.StringToWordVector;

public class NaiveBayesDemo {

    public static final String TRAINING_DATA_SET_FILENAME = "naive-train.arff";
    public static final String TESTING_DATA_SET_FILENAME = "naive-test.arff";
    public static final String PREDICTION_DATA_SET_FILENAME = "naive-confused.arff";

    public static Instances getDataSet(String fileName) throws Exception {

        StringToWordVector filter = new StringToWordVector();
            int classIdx = 1;
            ArffLoader loader = new ArffLoader();
             loader.setSource(NaiveBayesDemo.class.getResourceAsStream("/"+fileName));
            Instances dataSet = loader.getDataSet();

            dataSet.setClassIndex(classIdx);
            filter.setInputFormat(dataSet);
            dataSet = Filter.useFilter(dataSet, filter);
            return dataSet;
    }

    }

    public static void process() throws Exception {

            Instances trainingDataSet = getDataSet();
            Instances testingDataSet = getDataSet();
            Instances predictingDataSet = getDataSet();

            Classifier classifier = new NaiveBayesMultinomial();

            classifier.buildClassifier(trainingDataSet);

            Evaluation eval = new Evaluation(trainingDataSet);
            eval.evaluateModel(classifier, testingDataSet);
            /** Print the algorithm summary */
            System.out.println("** Naive Bayes Classifier **");
            System.out.println(eval.toSummaryString());
            System.out.print("Result");
            System.out.println(classifier);
            for (int i = 0; i < predictingDataSet.numInstances(); i++) {
                    System.out.println(predictingDataSet.instanse(i));
                    double index = classifier.classifyInstance(predictingDataSet.instance(i));
                    String className = trainingDataSet.attribute(0).value((int) index);
                    System.out.println(className);
            }


}
```

7.3 Listing algorithm development

Under listing algorithms, I am using back listing, whitelisting and greylisting to integrate and create a new algorithm. I designed the algorithm but haven’t created programmed yet. The whitelisting algorithm will allow all selected emails to communicate. The blacklisting algorithm will block all selected emails to communicate. The Greylisting algorithm is an intermediate between white and blacklisting. It will reject emails temporarily to check whether it's from a legitimate server. If its legitimate sender server will send it again.

8. Schedule

Project plan		
Stage	**Deadline**	**Outcome**
9. Background research	09/10/2019	Project proposal
10. Initiation	03/11/2019	Project initiation document
11. High-level design	12/11/2019	Design plan (DB schema, GUI style, Pseudo codes, Architecture)
12. Increment 1	18/12/2019	Front-end (algorithms)
13. Increment 2	01/02/2020	Back-end (user interfaces)
14. Increment 3	17/02/2020	The integrated backend and front-end
15. Testing and evaluation	20/03/2020	Assembled product and statistical evaluation
16. Finalizing	01/04/2020	Final product and report

9. Learning undertaken and required

Undertaken:

- Learned Java programming language.
- Learned C# programming language
- Learned HTML, CSS, Javascript, PHP, Perl
- Learned Microsoft Visual Studio IDE.
- Networking concepts
- Mathematical model of Naïve Bayesian theory and related concepts.

Required:

- More C# programming knowledge.
- More SQL database knowledge.
- Learn Apache Maven.
- Advanced C# programming concepts.
- Linux administration

19.6. Interim report 2

1. Purpose

This report details the processes of the system going to design, build, and test a Spam filtering application. The objective of this effort is to make a spam filtering application with a much lower false-positive rate. The outcome ought to be a client-side spam filter, that is easy to operate by user and robust in work. The principal requirements were first recognized to develop a good final product. In the first place, the program should bring about less inadvertently blocking real ham messages as its cost is a lot higher than giving spam a chance to bypass the security. I want to make our email inbox less annoying and more helpful.

2. Background

Nowadays, communication via email is very prominent. Everyone has an email ID, and everyone sends and receives emails. Spammers use bulk email technologies to send thousands of emails at once to random users. Most of the global email traffic is now spam, which means for one legitimate email you receive four spam emails. Such a huge amount of spam makes servers busy by processing spam and waste valuable resources. Organizations with 24 users or less receive 600 spam messages per month. Those unnecessary traffic cause many problems like reducing bandwidth, lowering service performances, and reducing the productivity of employees.

3. Motivation

I have a setup of four email IDs on my computer and mobile phone. Two of which are my personal emails and another two from NSBM University and Plymouth University. Every day when I look at the email box, I see at least 3 spam emails. Even though I know those are spam messages still I want to look at what are they and who had sent them, this creates distractions every time when I am going to check my emails. So, I thought about creating a spam filtering application for my final project. I may not be able to create a commercial level spam filter, but at least I want to give it a try to create a new technique to stop email spams. In the project, my idea is to Implement a demo system for a spam filtering application.

4. Goal

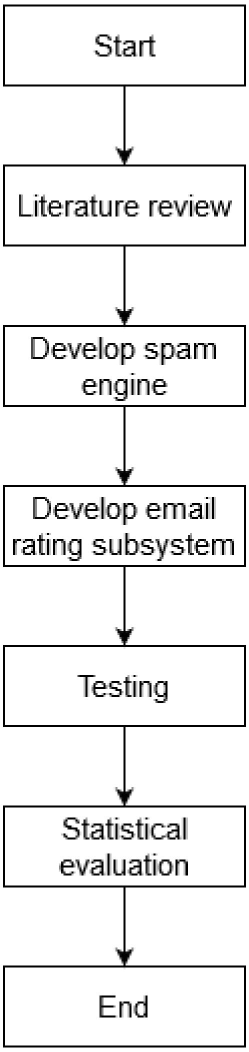
The goal of this project is to provide a new spam filter to server-side applications to enhance the spam filtering process with a new technique. This application will not be a replacement for your typical regular email application. My intent is to create a security technique for spam filtering not to create a spam filtering software.

5. System description

If an organization is using its own email server, they can use my email filter. I'm creating this spam filter for Zimbra open source mail server. The email rating subsystem which is a zimlet that can be integrated into the Zimbra mail server can be used to get feedback from the users. Users can provide feedback anytime about an email that they are spam or not spam. That data can be used as feedback to improve the spam filtering process in the future.

6. Project methodology

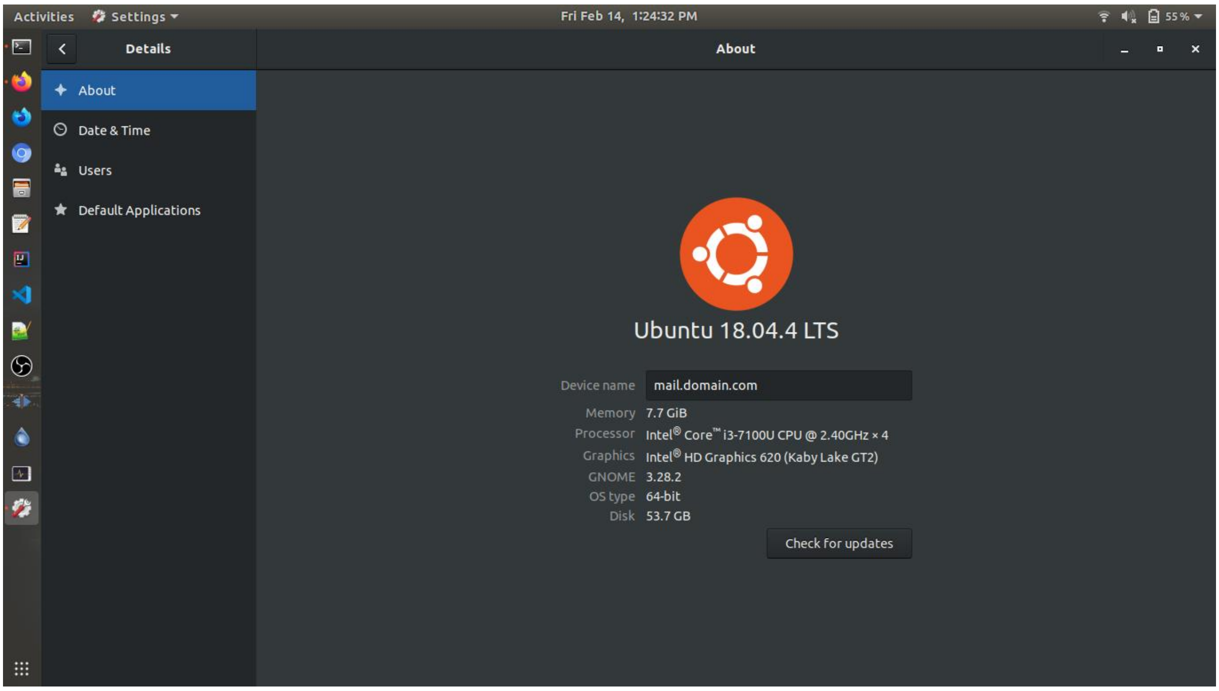
Spam filtering application consists of the main two important parts which are spam engine and email rating system. This whole development process involves six stages. Starting from the literature review which allows me to do background research and acquire knowledge about the project I'm going to do. Next, developing the spam engine, which is the core of the project, which is the toughest and most advanced phase, then developing an email rating subsystem, which is the easiest phase, next is data collection which allows collecting data to test, then data testing and statistical evaluation.



7. System development

7.1 Setup Linux environment

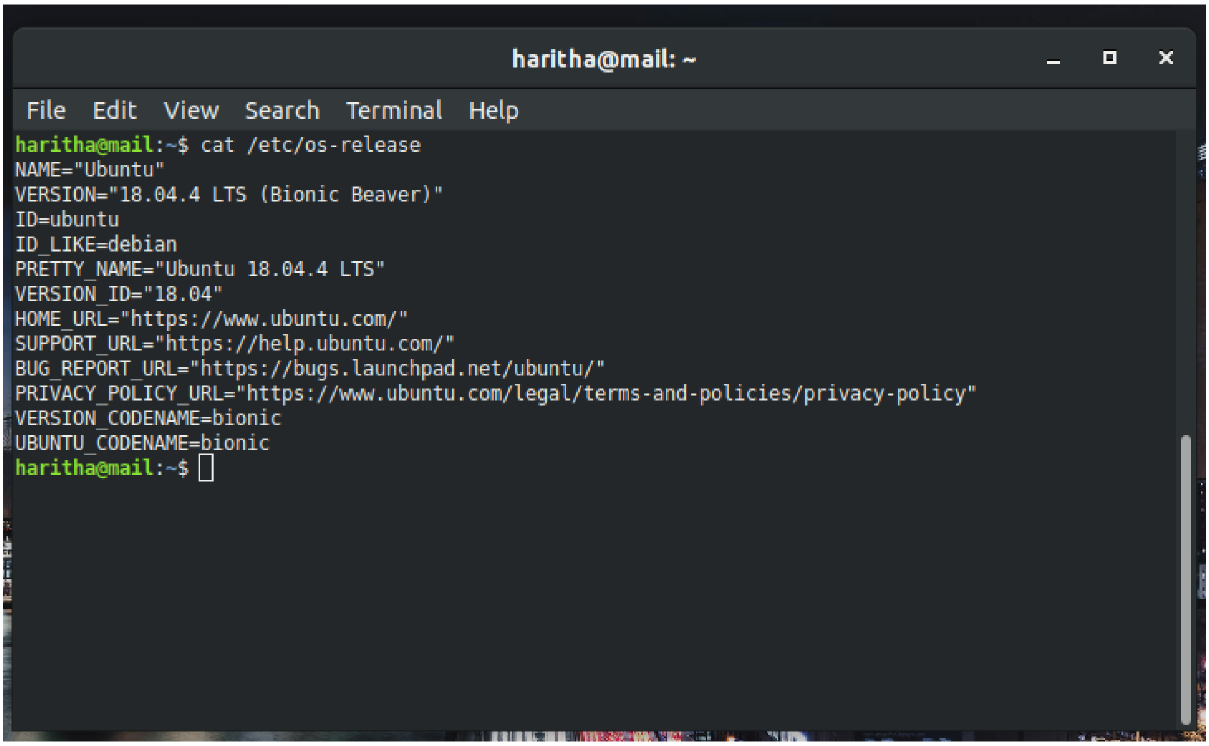
Ubuntu 18.04 LTS version of Linux is used to carry out the project. I have installed the Ubuntu operating system on my computer.



Ubuntu 18.04.4 LTS

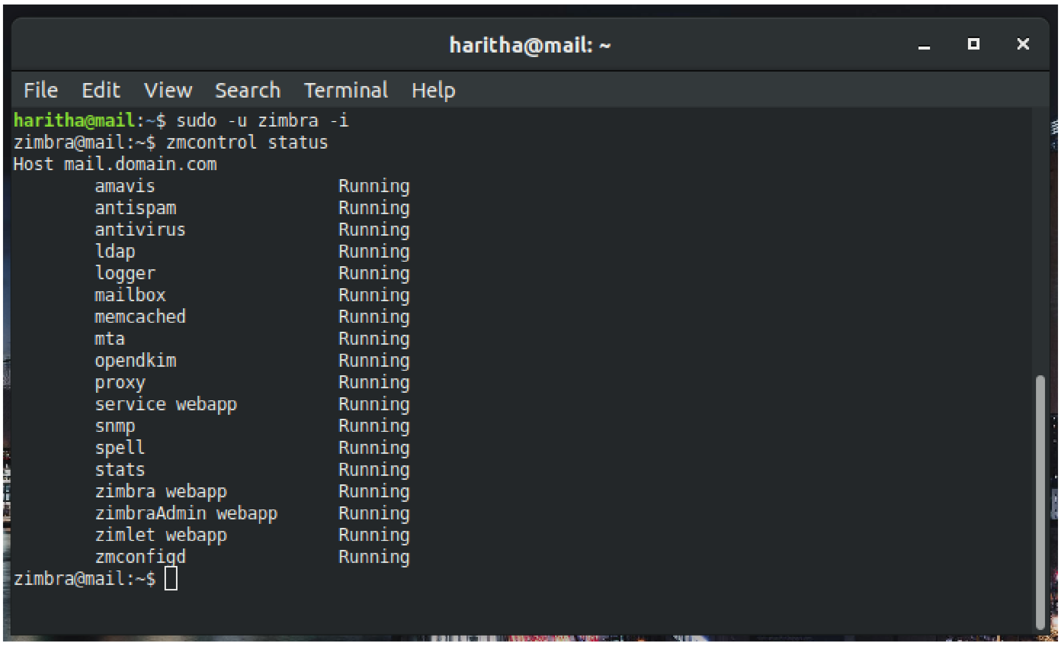
Device name mail.domain.com
Memory 7.7 GiB
Processor Intel® Core™ i3-7100U CPU @ 2.40GHz × 4
Graphics Intel® HD Graphics 620 (Kaby Lake GT2)
GNOME 3.28.2
OS type 64-bit
Disk 53.7 GB

7.2 Setup Zimbra email server

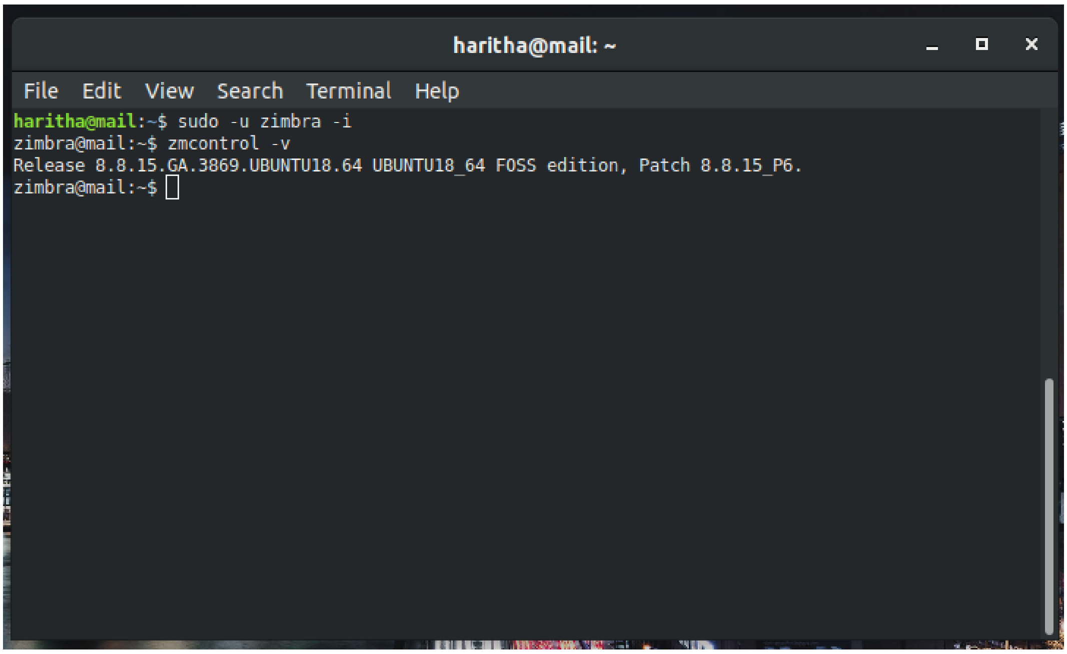


```
haritha@mail:~$ cat /etc/os-release
NAME="Ubuntu"
VERSION="18.04.4 LTS (Bionic Beaver)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 18.04.4 LTS"
VERSION_ID="18.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=bionic
UBUNTU_CODENAME=bionic
haritha@mail:~$ 
```

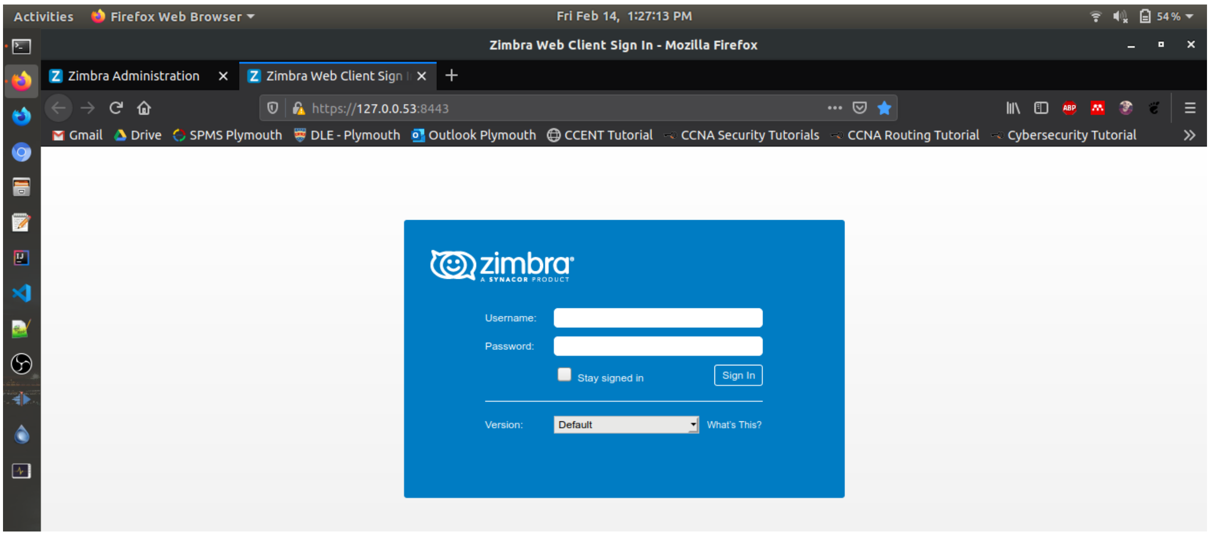
I am using Zimbra open-source mail server to do this project. Linux environment is used as the operating system. Ubuntu 18.04 LTS is the version of Linux distribution used. It only supports Ubuntu 14.04, 16.04 and 18.04 versions of ubuntu. Other supporting platforms like CentOS and Fedora were not used because I am only familiar with Ubuntu distro. The Zimbra mail server can use to set up all the server-side functionalities needed to send and receive emails. Functionalities included are adding, deleting and managing user accounts, spam filtering, chat boxes, send an email, receive emails etc.

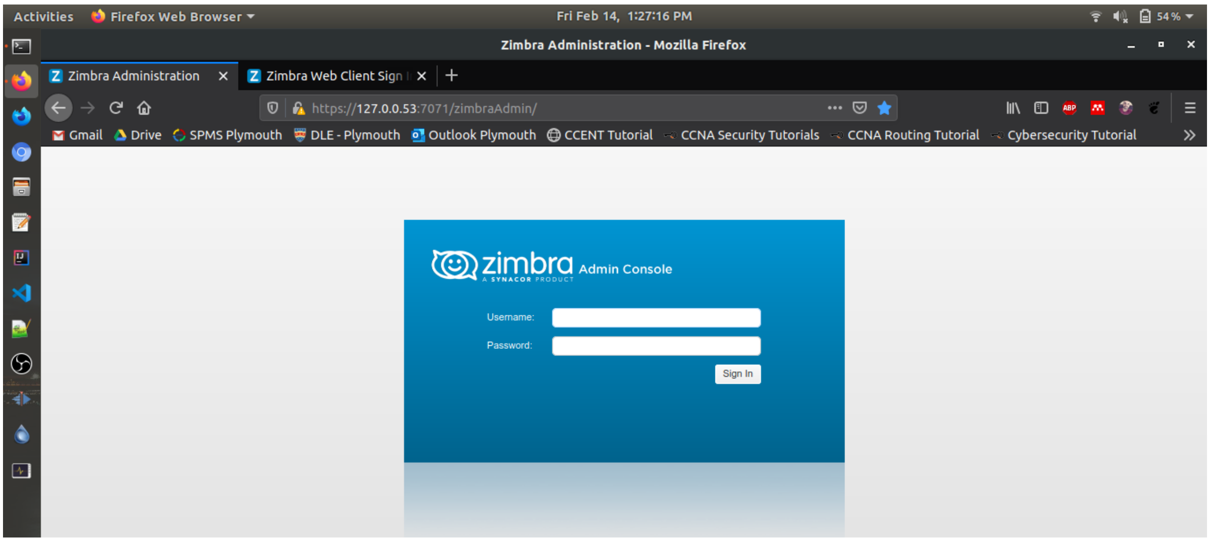


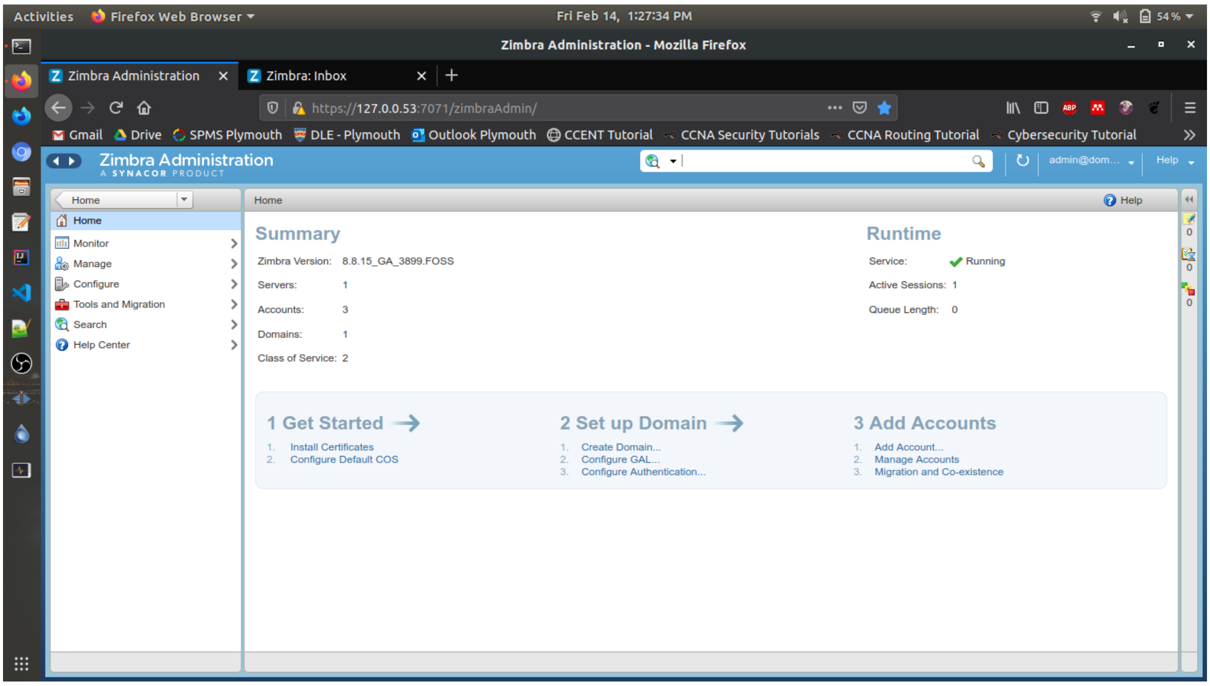
```
haritha@mail:~$ sudo -u zimbra -i
zimbra@mail:~$ zmcontrol status
Host mail.domain.com
        amavis                  Running
        antispam                Running
        antivirus               Running
        ldap                    Running
        logger                  Running
        mailbox                 Running
        memcached               Running
        mta                     Running
        opendkim                Running
        proxy                   Running
        service webapp          Running
        snmp                    Running
        spell                   Running
        stats                   Running
        zimbra webapp           Running
        zimbraAdmin webapp      Running
        zimlet webapp           Running
        zmconfigd               Running
zimbra@mail:~$ 
```



```
haritha@mail:~$ sudo -u zimbra -i
zimbra@mail:~$ zmcontrol -v
Release 8.8.15.GA.3869.UBUNTU18.64 UBUNTU18_64 FOSS edition, Patch 8.8.15_P6.
zimbra@mail:~$ 
```


Activities
Firefox Web Browser
Fri Feb 14, 1:27:13 PM
Zimbra Web Client Sign In - Mozilla Firefox
Zimbra Administration
Zimbra Web Client Sign I
https://127.0.0.53:8443
Gmail
Drive
SPMS Plymouth
DLE - Plymouth
Outlook Plymouth
CCENT Tutorial
CCNA Security Tutorials
CCNA Routing Tutorial
Cybersecurity Tutorial
zimbra
A SYNACOR PRODUCT
Username:
Password:
Stay signed in
Sign In
Version:
Default
What's This?


Activities
Firefox Web Browser
Fri Feb 14, 1:27:16 PM
Zimbra Administration - Mozilla Firefox
Zimbra Administration
Zimbra Web Client Sign I
https://127.0.0.53:7071/zimbraAdmin/
Gmail
Drive
SPMS Plymouth
DLE - Plymouth
Outlook Plymouth
CCENT Tutorial
CCNA Security Tutorials
CCNA Routing Tutorial
Cybersecurity Tutorial
zimbra Admin Console
A SYNACOR PRODUCT
Username:
Password:
Sign In



Summary

Zimbra Version: 8.8.15_GA_3899.FOSS

Servers: 1

Accounts: 3

Domains: 1

Class of Service: 2

Runtime

Service: ✓ Running

Active Sessions: 1

Queue Length: 0

1 Get Started →

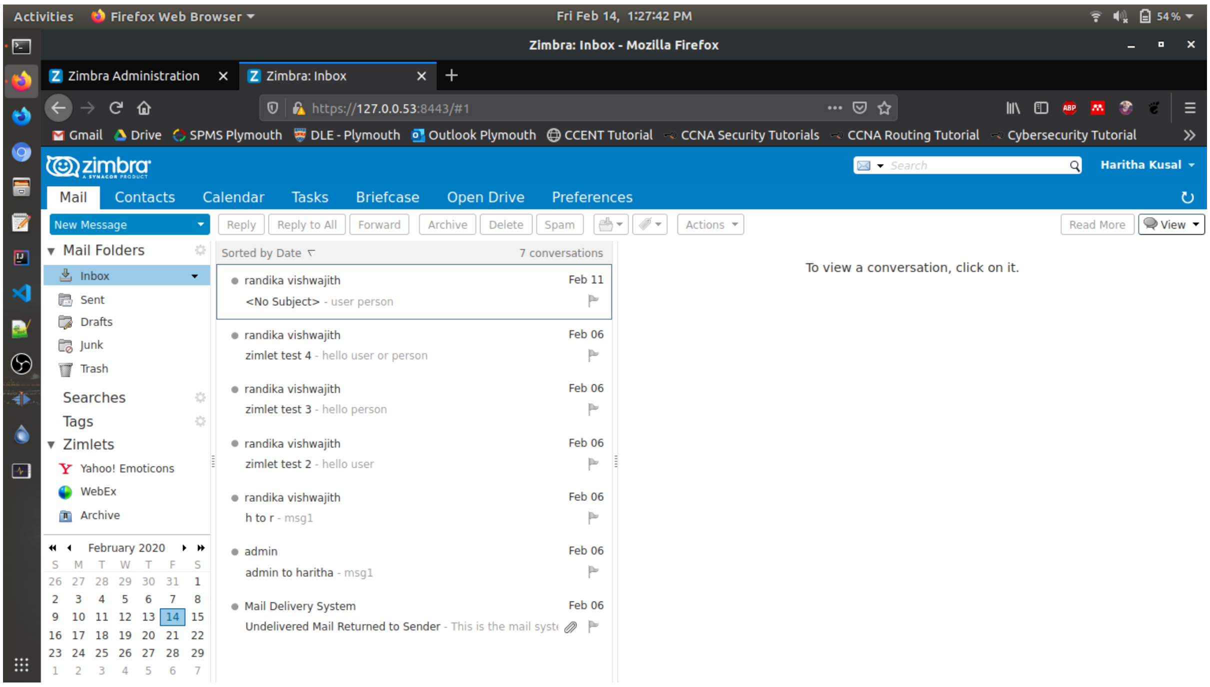
1. Install Certificates
2. Configure Default COS

2 Set up Domain →

1. Create Domain...
2. Configure GAL...
3. Configure Authentication...

3 Add Accounts

1. Add Account...
2. Manage Accounts
3. Migration and Co-existence



Sorted by Date 7 conversations

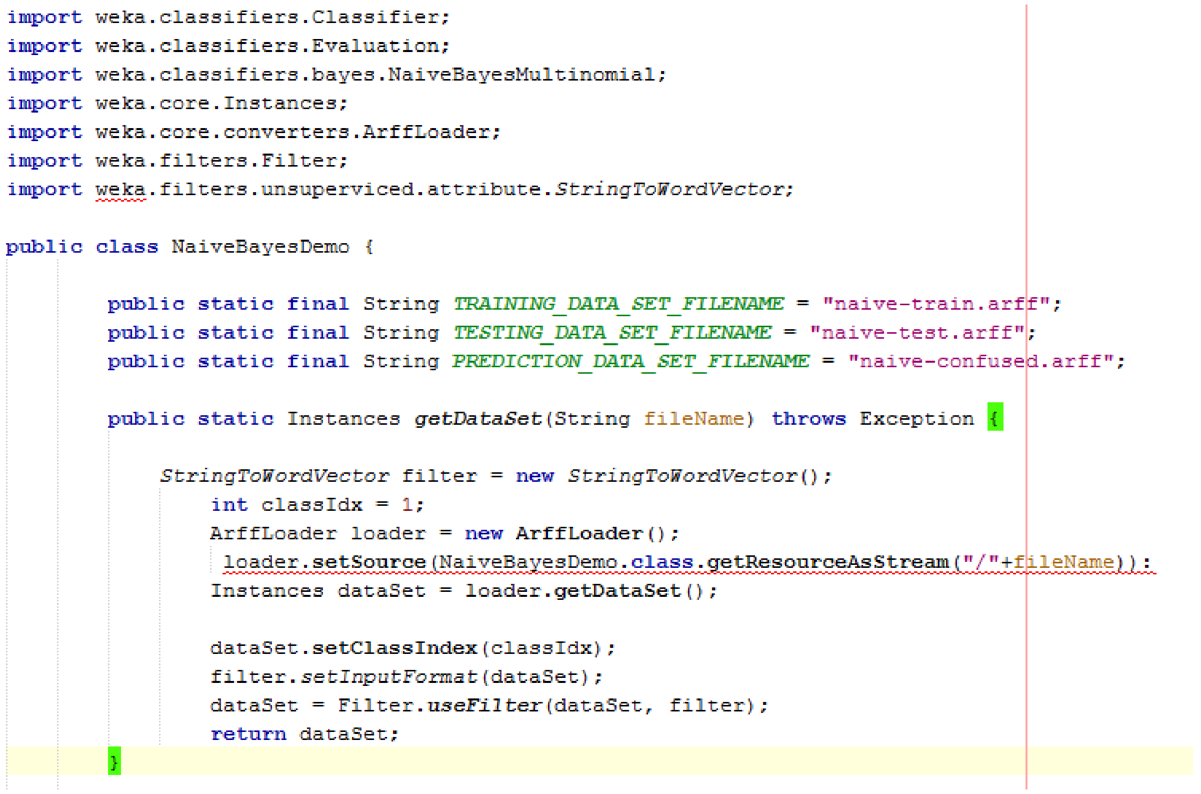
randika vishwajith <No Subject> - user person	Feb 11
randika vishwajith zimlet test 4 - hello user or person	Feb 06
randika vishwajith zimlet test 3 - hello person	Feb 06
randika vishwajith zimlet test 2 - hello user	Feb 06
randika vishwajith h to r - msg1	Feb 06
admin admin to haritha - msg1	Feb 06
Mail Delivery System Undelivered Mail Returned to Sender - This is the mail syst	Feb 06

To view a conversation, click on it.

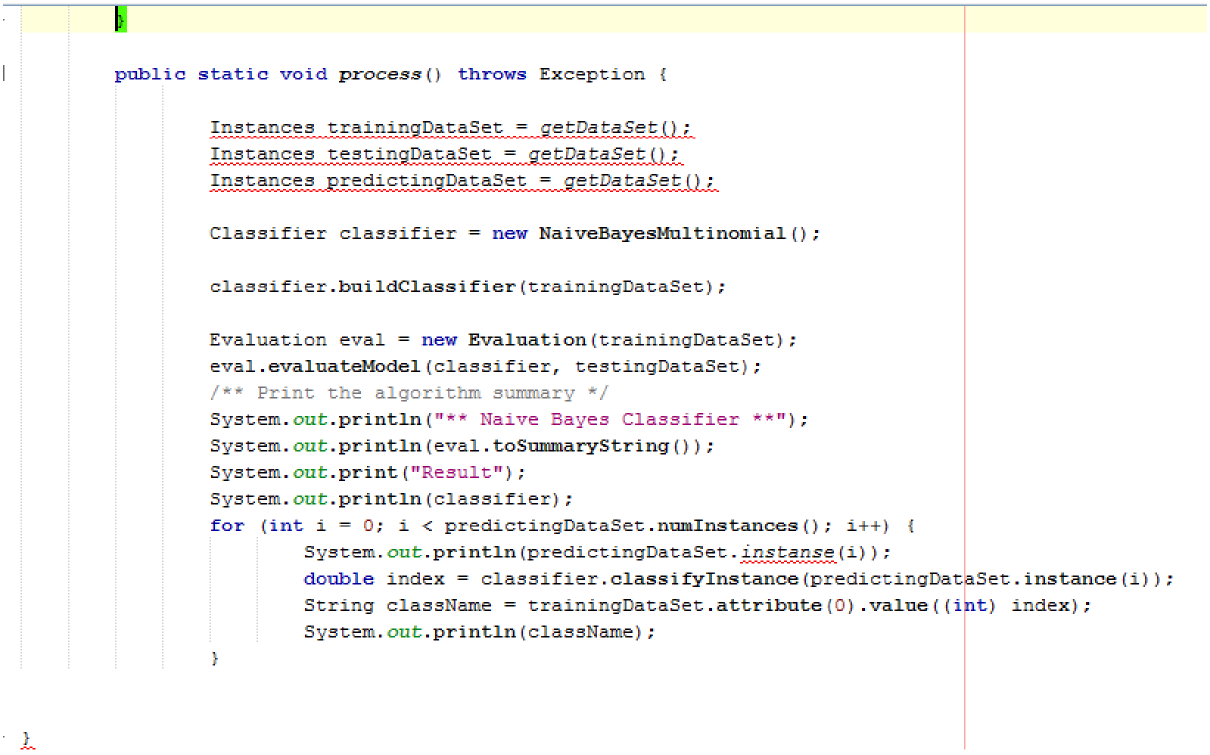
7.3 Spam engine

Spam engine is a part of the Zimbra mail server which can be integrated as a zimlet to the server. A Zimlet is a subsystem that can be used to extend the functionalities of the server. Apache Netbeans is used as the IDE to develop spam engine java code.

Below shows a few of the codes programmed using Java:



```java
import weka.classifiers.Classifier;
import weka.classifiers.Evaluation;
import weka.classifiers.bayes.NaiveBayesMultinomial;
import weka.core.Instances;
import weka.core.converters.ArffLoader;
import weka.filters.Filter;
import weka.filters.unsuperviced.attribute.StringToWordVector;

public class NaiveBayesDemo {

        public static final String TRAINING_DATA_SET_FILENAME = "naive-train.arff";
        public static final String TESTING_DATA_SET_FILENAME = "naive-test.arff";
        public static final String PREDICTION_DATA_SET_FILENAME = "naive-confused.arff";

        public static Instances getDataSet(String fileName) throws Exception {

            StringToWordVector filter = new StringToWordVector();
                int classIdx = 1;
                ArffLoader loader = new ArffLoader();
                 loader.setSource(NaiveBayesDemo.class.getResourceAsStream("/"+fileName));
                Instances dataSet = loader.getDataSet();

                dataSet.setClassIndex(classIdx);
                filter.setInputFormat(dataSet);
                dataSet = Filter.useFilter(dataSet, filter);
                return dataSet;
        }
```



```java
        }

        public static void process() throws Exception {

                Instances trainingDataSet = getDataSet();
                Instances testingDataSet = getDataSet();
                Instances predictingDataSet = getDataSet();

                Classifier classifier = new NaiveBayesMultinomial();

                classifier.buildClassifier(trainingDataSet);

                Evaluation eval = new Evaluation(trainingDataSet);
                eval.evaluateModel(classifier, testingDataSet);
                /** Print the algorithm summary */
                System.out.println("** Naive Bayes Classifier **");
                System.out.println(eval.toSummaryString());
                System.out.print("Result");
                System.out.println(classifier);
                for (int i = 0; i < predictingDataSet.numInstances(); i++) {
                        System.out.println(predictingDataSet.instanse(i));
                        double index = classifier.classifyInstance(predictingDataSet.instance(i));
                        String className = trainingDataSet.attribute(0).value((int) index);
                        System.out.println(className);
                }


 }
```

8. Schedule

Project plan		
Stage	**Deadline**	**Outcome**
17. Background research	09/10/2019	Project proposal
18. Initiation	03/11/2019	Project initiation document, Define the scope of the project.
19. High level design	12/11/2019	Design plan (DB schema, GUI style, Pseudo codes, Architecture etc)
20. Increment 1	01/01/2020	Install Ubuntu Linux distro and setup Zimbra email server.
21. Increment 2	01/02/2020	Develop spam engine
22. Increment 3	01/03/2020	Develop an email rating system
23. Testing and evaluation	20/03/2020	Assembled product and statistical evaluation
24. Finalizing	07/04/2020	Final product and report

Learning undertaken and required

- Java programming language.
- Linux environment installation and configurations.
- Shell scripting.
- Javascript scripting language.
- Zimbra, iRedmail server setup, configurations and
- Spam filtering theories and related concepts.
- MySQL database knowledge.
- Apache Netbeans IDE/Intellij IDEA Community IDE.

9. Table of content (Draft)

10. Introduction (Draft)

In 2018, approximately 281 billion (*Daily number of e-mails worldwide 2023 | Statista*, 2019) e-mails were sent and received every day worldwide. This figure is projected to increase to over 347 billion (*Daily number of e-mails worldwide 2023 | Statista*, 2019) daily e-mails in 2023. Spam has also grown gradually with the development of the internet and now 75% - 80% (Blanzieri and Bryl, 2008) of global email traffic is spam. Spam emails are useless, junk messages arrive into users' email inbox spontaneously and sent in bulk by the spammer. The main reason why people don't like spam is it's annoying to receive unknown and unwanted messages continuously during the day. Spam makes our email inboxes full instantly and reduces work efficiency and productivity. Users had to always keep sorting out legitimate messages from unwanted messages. Most spam messages have a commercial purpose. People who want to promote their commercial products send advertisements in bulk to spontaneous email IDs. Some malicious personals use spam messages to spread malware and links that lead to suspicious websites to gain private data of users.

A community is a social structure that shares personal values, cultural values, business goals, attitudes, or a world view. Frequently people in those communities work together to achieve important tasks. Those tasks can be varying from cleaning up the neighborhood to the takedown of governments and empires. An online community is connected by offering and accepting. Community is affinity, identity, and kinship that make room for ideas, thoughts, and solutions. In general, online communities value knowledge and information as currency or social resources. The difference between online communities from their physical counterparts is the extent and impact of "weak ties," which are the relationships acquaintances or strangers form to acquire information through online networks. Relationships among members in a virtual community tend to focus on information exchange, entertainment, professional, and sports virtual groups focused their activities on obtaining information(Horrigan, Rainie and S, 2001).

The aim of this project is to develop an anti-spam solution that will be based on community ratings to increase the accuracy of spam detection. Community ratings are very valuable. Because the ones who give feedbacks are humans. No algorithm can equate with a human who is using his previous experiences in the real-world to decide something. A large set of data given directly by users is always better than an algorithm that processes sample data. It will further enhance spam filters by providing real data from real human experience because spam filters always rely on predefined algorithms that work in the same manner repetitively. Community rating will fill a gap within automation. Spam filters can personalize themselves using given feedbacks to increase accuracy to reduce false positives and false negatives.

11. Scope (Draft)

This project is a community-powered solution to spam filtering. It will consist of a spam filtering engine that will further enhance the accuracy of spam detection by using the input of user ratings. Often spam filters miss some of the spam messages and deliver them to inbox. In order to catch those spam messages, the system needs to increase the sensitivity of algorithms running inside the spam filters.

Usually, a community refers to a group of people living in a common geographical location. A certain community consists of thousands if not millions of people. People already do a lot of individual problems solving, and there's a good deal of merit in that. But many of the problems and challenges people face as individuals or as members belong to an organization affect everyone in the group. It makes sense then, that everyone being a contributor to a solution is very effective. Many things can be accomplished with a community full of people helping to solve a problem. The ability to collect such a large amount of data from a community without utilizing any extra effort is groundbreaking. People’s opinions about the experience they have with something is helpful information that can use to adjust that thing to fit their needs more accurately.

The spam filter will detect, and filter emails using spam filtering techniques and as a new technique, the system will request a rating from users to rate received spam emails missed by the spam filter. Users should consider the content of the email and decide If they are spam or not spam according to their own point of view. This technique will also help to personalize spam according to the user requirement. A spam email for someone may not be spam for another person. Therefore, individual feedback can also be used to meet individual needs while filtering messages.

12. Background (Draft)

12.1 What is email?

Email also known as electronic mail or e-mail is a technique used to transfer messages between two electronic devices. More clearly, it is a text, photo, audio or video message transferring method between one or more individuals at once located in remote areas in different geographical locations via a computer network.

In 1971, Ray Tomlinson sent the world’s first email to himself containing the message “something like QWERTYUIOP” using ARPANET. The Advanced Research Project Agency Network also known as ARPANET is the initial version of the modern internet used by the military of the United States of America. Email uses a centralized system which is also called an email server to transfer and store emails. Email server act as an intermediate between the sender and receiver of the email to facilitate and allow smooth transfer without any disruptions. In present, email sent by mobile holds 43 percent of e-mail opens (*Global email platform market share 2018 | Statista*, 2018). Desktop e-mail clients’ open share had declined to 18 percent, and webmail accounted for 39 percent of opens. Based on the dominance of mobile, it is no surprise that the iPhone e-mail app was the most popular e-mail client, accounting for 29% of e-mail opens (*Global email platform market share 2018 | Statista*, 2018). Gmail was ranked second with a 27% open share. Gmail is a free e-mail service owned by Google, and the company reported 1.5 billion (*Number of Gmail active users 2018 | Statista*, 2018) active Gmail users worldwide in October 2018.

Email gives a user many advantages over traditional post mail. Anyone can send an email virtually without costing any money. The user only needs a working internet connection and an email account to send an email. Email can reach almost everywhere if that location has internet connectivity in an instant. Transferring multiple audios, videos, texts, and files is just a matter of user requirement. Environmental friendliness and ability for long-term storage are features totally ahead of traditional post mail.

12.2 What is email spam?

"Spam is the use of electronic messaging systems (including most broadcast media, digital delivery systems) to send unsolicited bulk messages indiscriminately" (Nazirova, 2011). History of email spam goes to the mid-1900s when the internet became a reality. In 1978, Gary Turk sent the first modern spam message to 400 people using ARPANET. Nowadays, everyone with an email account knows what is a spam message? Mail Abuse Prevention System (MAPS) is the first technology to defend against spam messages. It used blacklisting senders' IP addresses to prevent future spam messages.

Spamming is a profitable business in today's world. Most spammers spam to earn money by sending out advertisements to random email IDs to promote someone's business products. The reason why there are so many spam messages is that it is so easy to create and spread. Anyone could easily become a spammer.

Spam now contains malware too. Once clicked by a user a malicious program will infect the system to steal private data or destroy information assets. Spam emails, when received in large numbers, overwhelm users' mail inboxes, making it harder to find important messages and information. This leads to a decrease in overall productivity, efficiency, and loss of valuable time. More importantly, spam messages are the most common mediums used to perform phishing which is the initial step for executing large-scale cyberattacks. Phishing, if done successfully, can lead to devastating cyber threats like malware attacks and even ransomware attacks. Spam messages cause damages to the system and user at different levels. Spam consumes network bandwidth resulting in reduced work network efficiency and financial damage to the internet service provider or user. Even today for a normal user two or three spam messages received to the user inbox daily. Users should manually delete them. This causes a waste of time and a reduction in work productivity. Accidental loss of legitimate emails while deleting spam results in financial loss or time wastes to users or organizations.

Some spam messages carrying malware may cause catastrophic damages to the whole IT system and organization if an employee opens that email. Sometimes the business may have to file for bankruptcy and face lawsuits if the company is regulated under the law of the European Union.

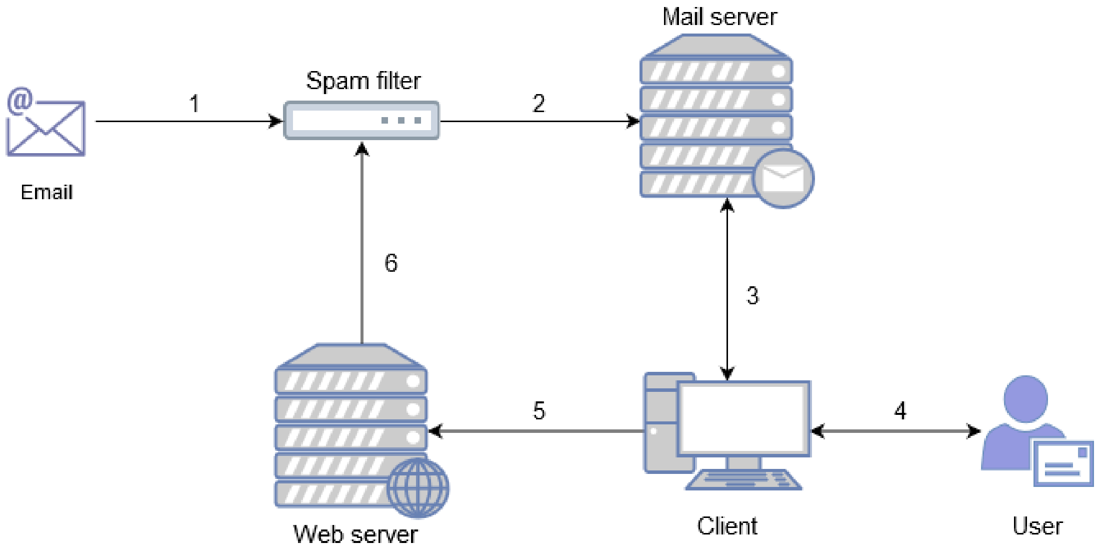
13. Literature review (Draft)

In September 2019, spam emails accounted for 54.68% of global email traffic (*Spam statistics: spam e-mail traffic share 2019 | Statista*, 2019). In that year China is responsible for generating 20.43% of global spam traffic while the United States is responsible for 13.37% (*Spam e-mail: countries of origin 2019 | Statista*, 2019). Global annual spam traffic was 69% in 2012 and it had decreased to 55% in 2018 and 53.68% in 2019 (*E-mail spam rate worldwide 2018 | Statista*, 2018). In 2018, the Federal Bureau of Investigation (FBI) of the United States stated in a report that financial loss due to business email compromise and email spam is USD 12.5 billion worldwide (*Internet Crime Complaint Center (IC3) | Business E-mail Compromise The 12 Billion Dollar Scam*, 2018).

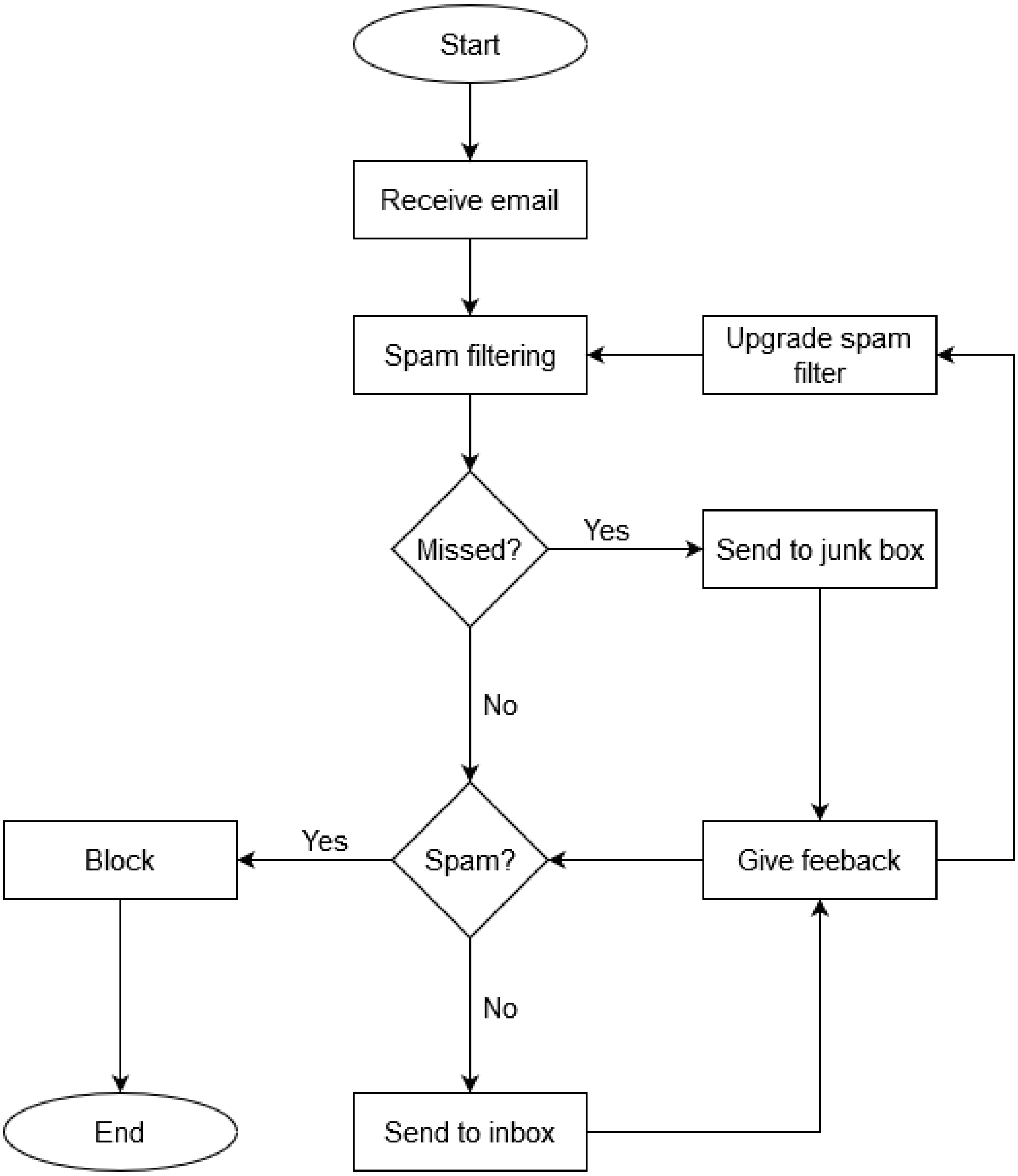
Current defense strategies in fighting email spam are much more sophisticated than a decade ago. The research focused on old everyday anti-spam techniques (Christina, Karpagavalli and Suganya, 2010) and modern approaches like machine learning (Bhowmick and Hazarika, 2016) have improved success rates of spam detection to virtually 100%. Old rule-based methods easy to deploy, configure and hard to maintain are becoming less efficient and producing more false positives.

14. Diagrams (Draft)

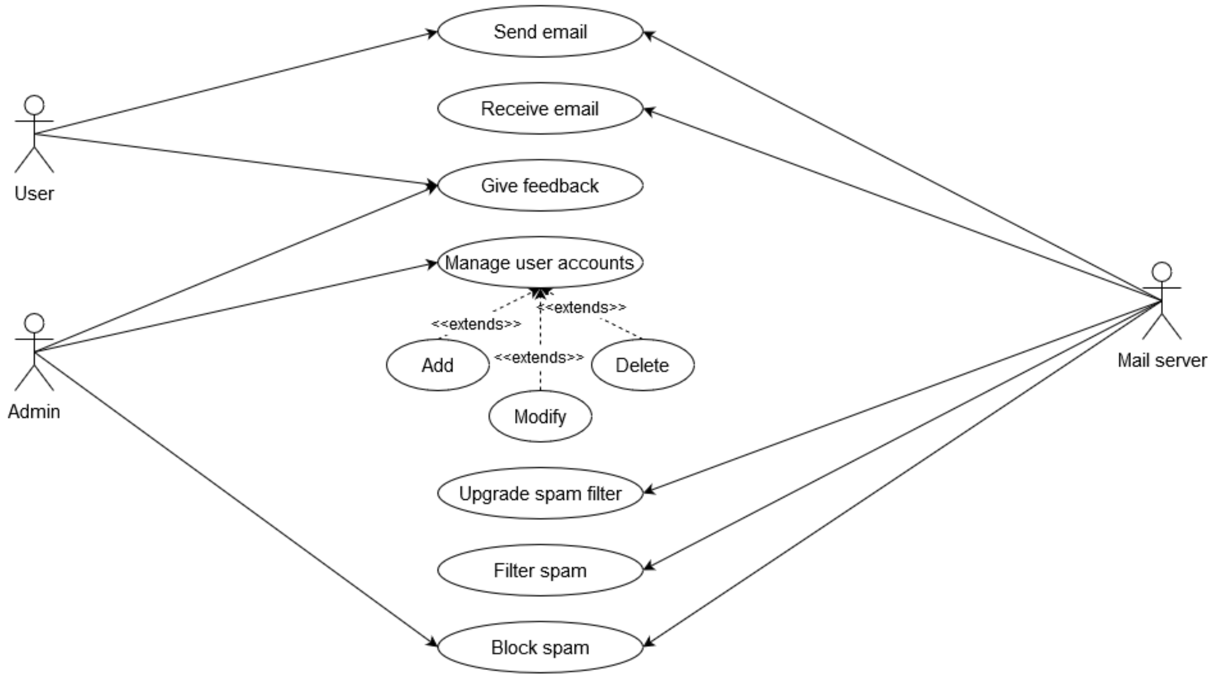
14.1 Basic system design (Draft)



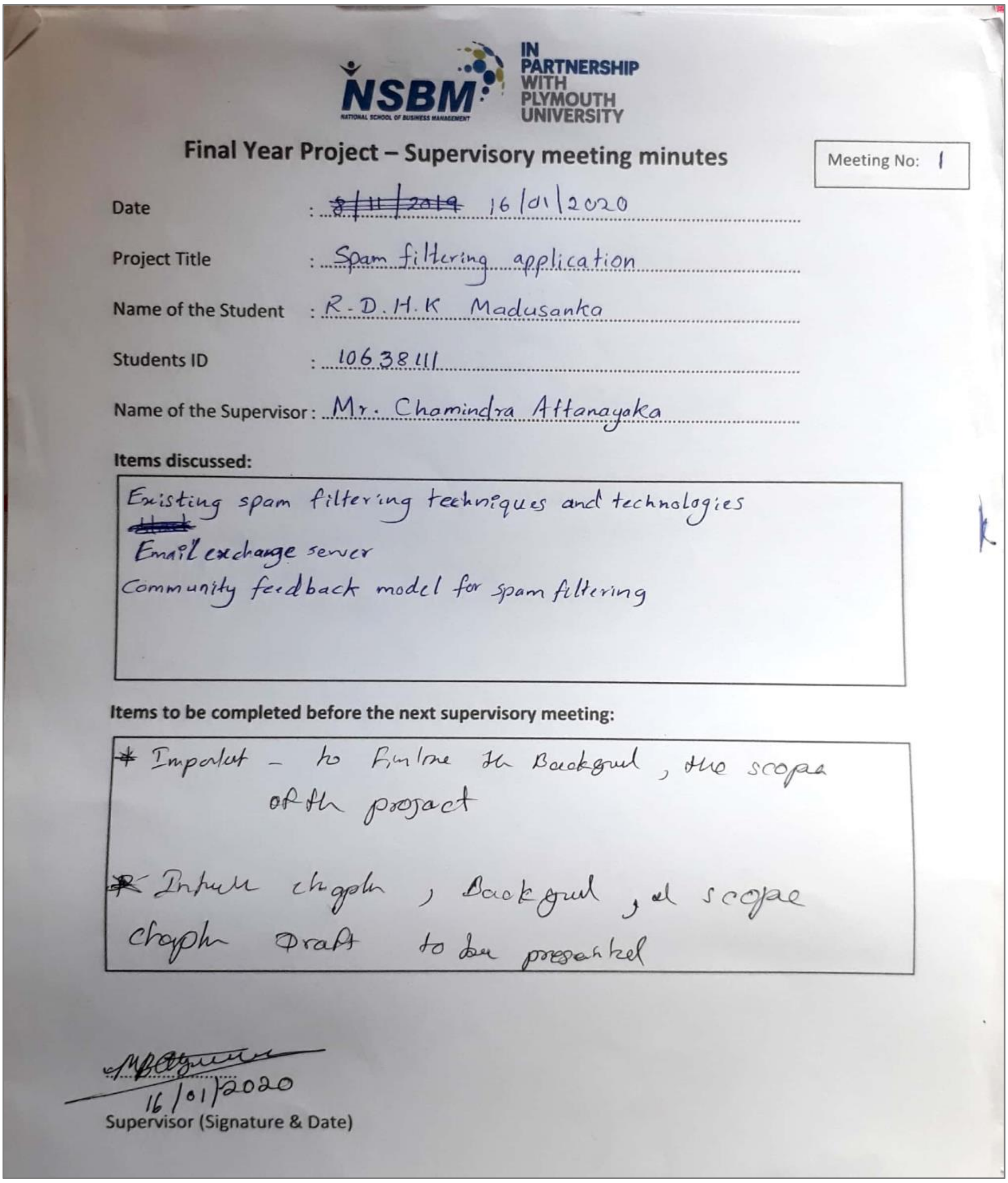
14.2 Email flow chart (Draft)



14.3 Use case diagram (Draft)



19.7. Supervisory meeting minutes



NSBM NATIONAL SCHOOL OF BUSINESS MANAGEMENT IN PARTNERSHIP WITH PLYMOUTH UNIVERSITY

Final Year Project – Supervisory meeting minutes

Meeting No: 1

Date : ~~8/11/2019~~ 16/01/2020

Project Title : Spam filtering application

Name of the Student : R.D.H.K Madusanka

Students ID : 10638111

Name of the Supervisor : Mr. Chamindra Attanayaka

Items discussed:

Existing spam filtering techniques and technologies

Email exchange server

Community feedback model for spam filtering

Items to be completed before the next supervisory meeting:

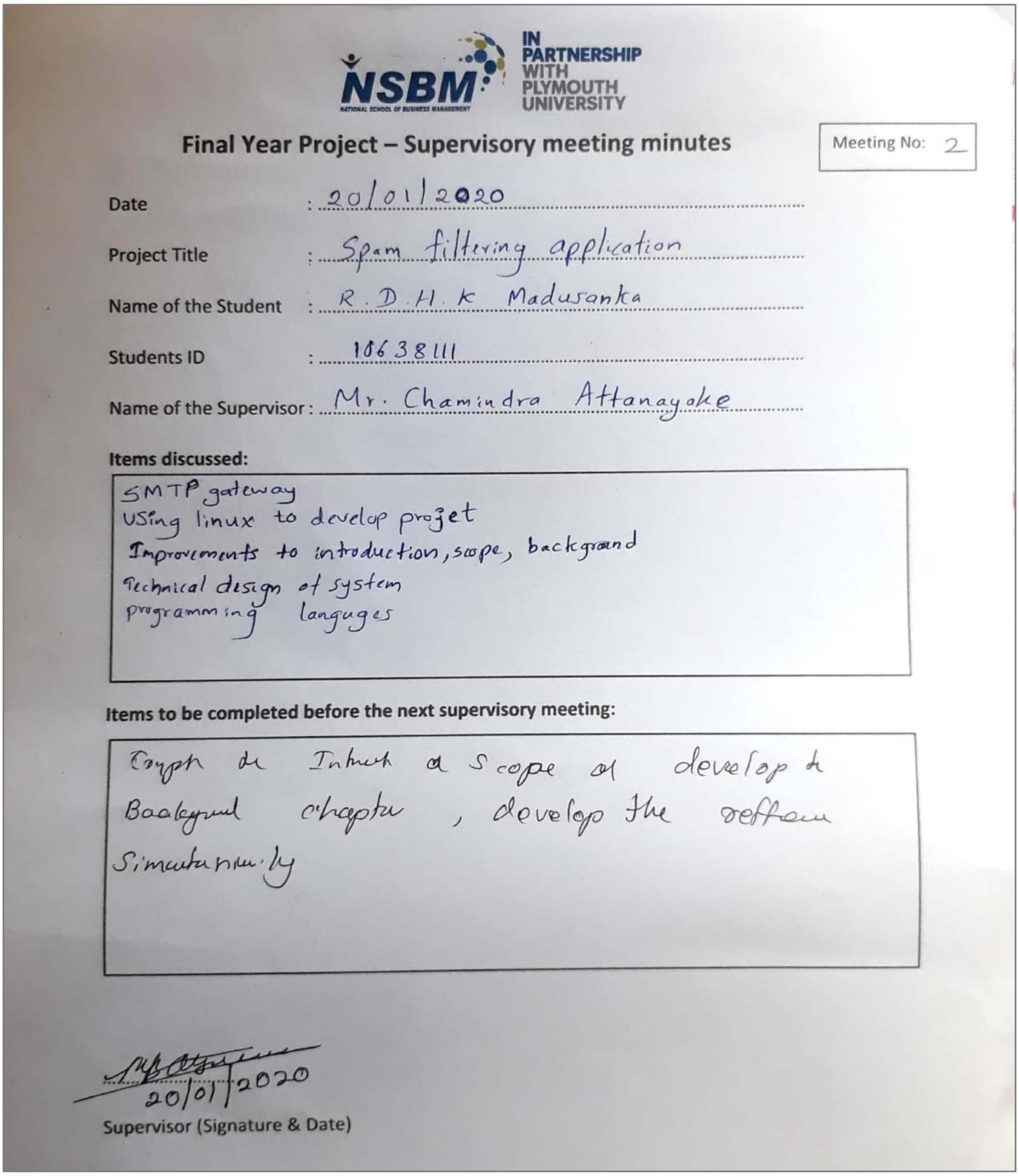
* Important – to finalise the Background, the scope of the project

* Introduction chapter, Background, and scope chapter Draft to be presented

16/01/2020

Supervisor (Signature & Date)

Figure 35: Meeting minutes 1



NSBM NATIONAL SCHOOL OF BUSINESS MANAGEMENT IN PARTNERSHIP WITH PLYMOUTH UNIVERSITY

Final Year Project – Supervisory meeting minutes

Meeting No: 2

Date : 20/01/2020

Project Title : Spam filtering application

Name of the Student : R. D. H. K Madusanka

Students ID : 10638111

Name of the Supervisor : Mr. Chamindra Attanayake

Items discussed:

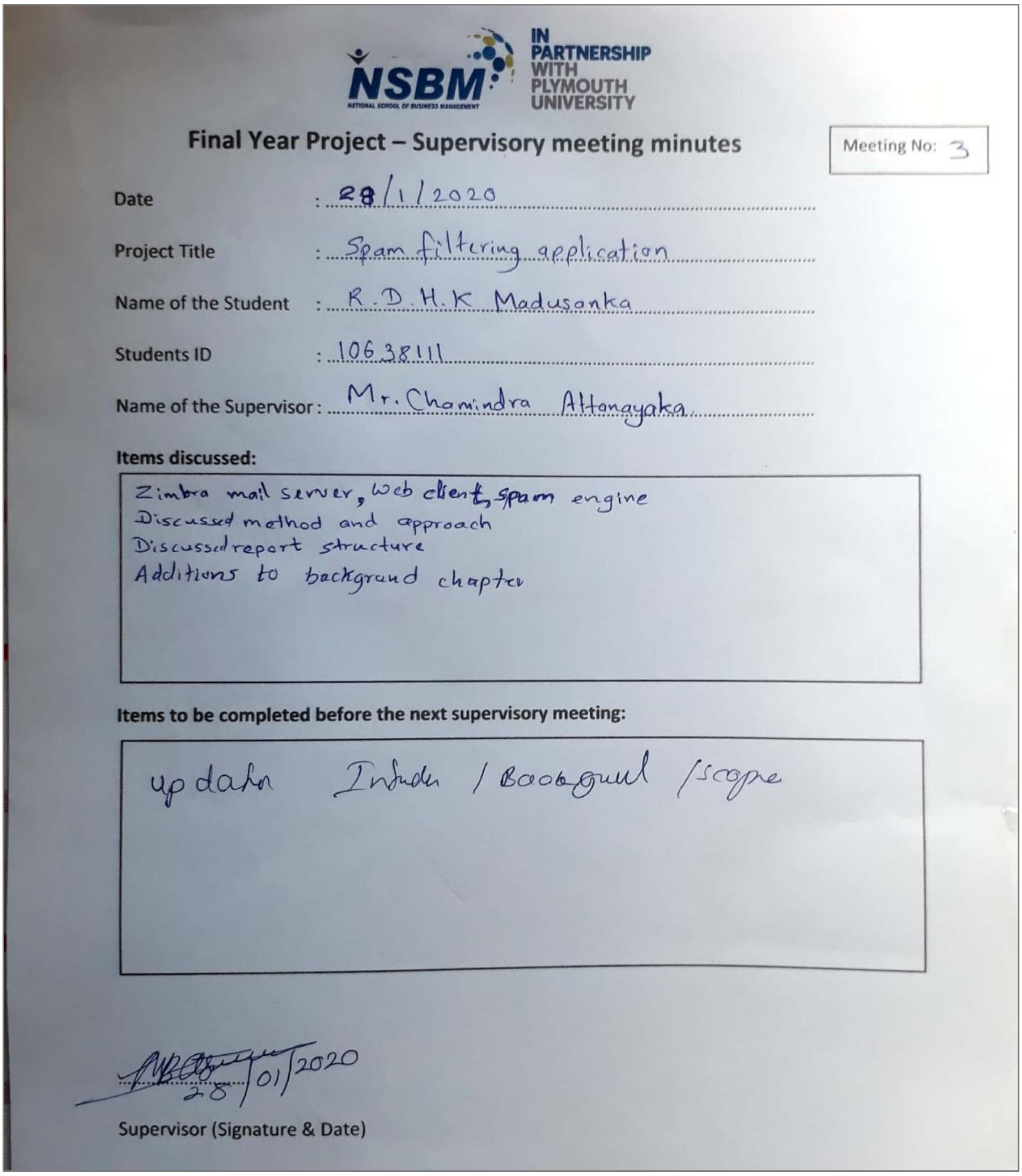
SMTP gateway
Using linux to develop projet
Improvements to introduction, scope, background
Technical design of system
programming languges

Items to be completed before the next supervisory meeting:

Complete the Introduct & Scope and develop the Background chapter, develop the software Simultaneously

20/01/2020
Supervisor (Signature & Date)

Figure 36: Meeting minutes 2



NSBM NATIONAL SCHOOL OF BUSINESS MANAGEMENT IN PARTNERSHIP WITH PLYMOUTH UNIVERSITY

Final Year Project – Supervisory meeting minutes

Meeting No: 3

Date : 28/1/2020

Project Title : Spam filtering application

Name of the Student : R.D.H.K Madusanka

Students ID : 10638111

Name of the Supervisor : Mr. Chamindra Attanayaka

Items discussed:

Zimbra mail server, Web client, spam engine
Discussed method and approach
Discussedreport structure
Additions to backgrund chapter

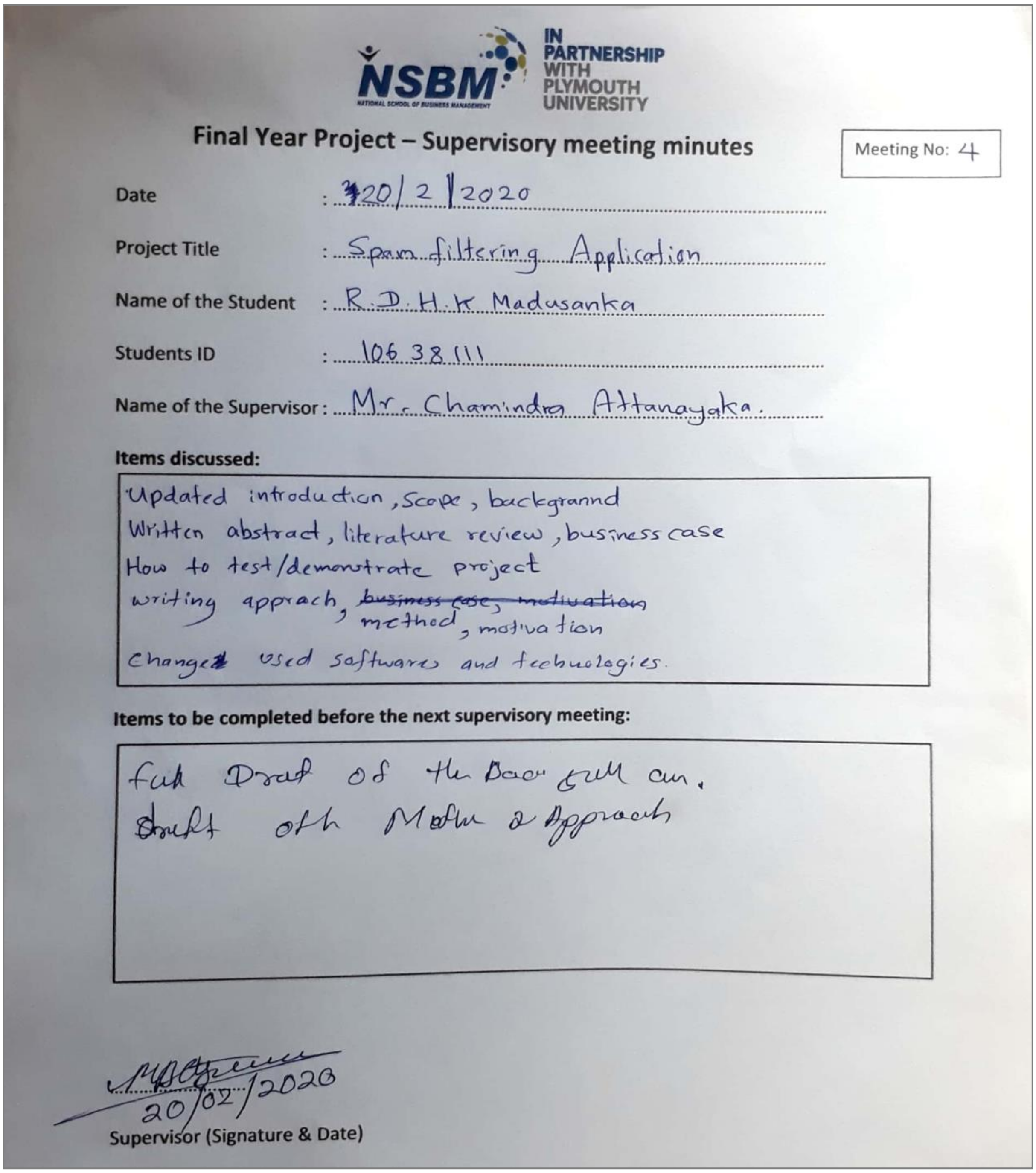
Items to be completed before the next supervisory meeting:

update Introdn / Background /scope

28/01/2020

Supervisor (Signature & Date)

Figure 37: Meeting minutes 3



NSBM NATIONAL SCHOOL OF BUSINESS MANAGEMENT IN PARTNERSHIP WITH PLYMOUTH UNIVERSITY

Final Year Project – Supervisory meeting minutes

Meeting No: 4

Date : 20/2/2020

Project Title : Spam filtering Application

Name of the Student : R.D.H.K Madusanka

Students ID : 10638111

Name of the Supervisor : Mr. Chamindra Attanayaka.

Items discussed:

Updated introduction, Scope, backgrannd
Written abstract, literature review, business case
How to test/demonstrate project
writing appraach, ~~business case, motivation~~ method, motivation
Changed used softwares and technologies.

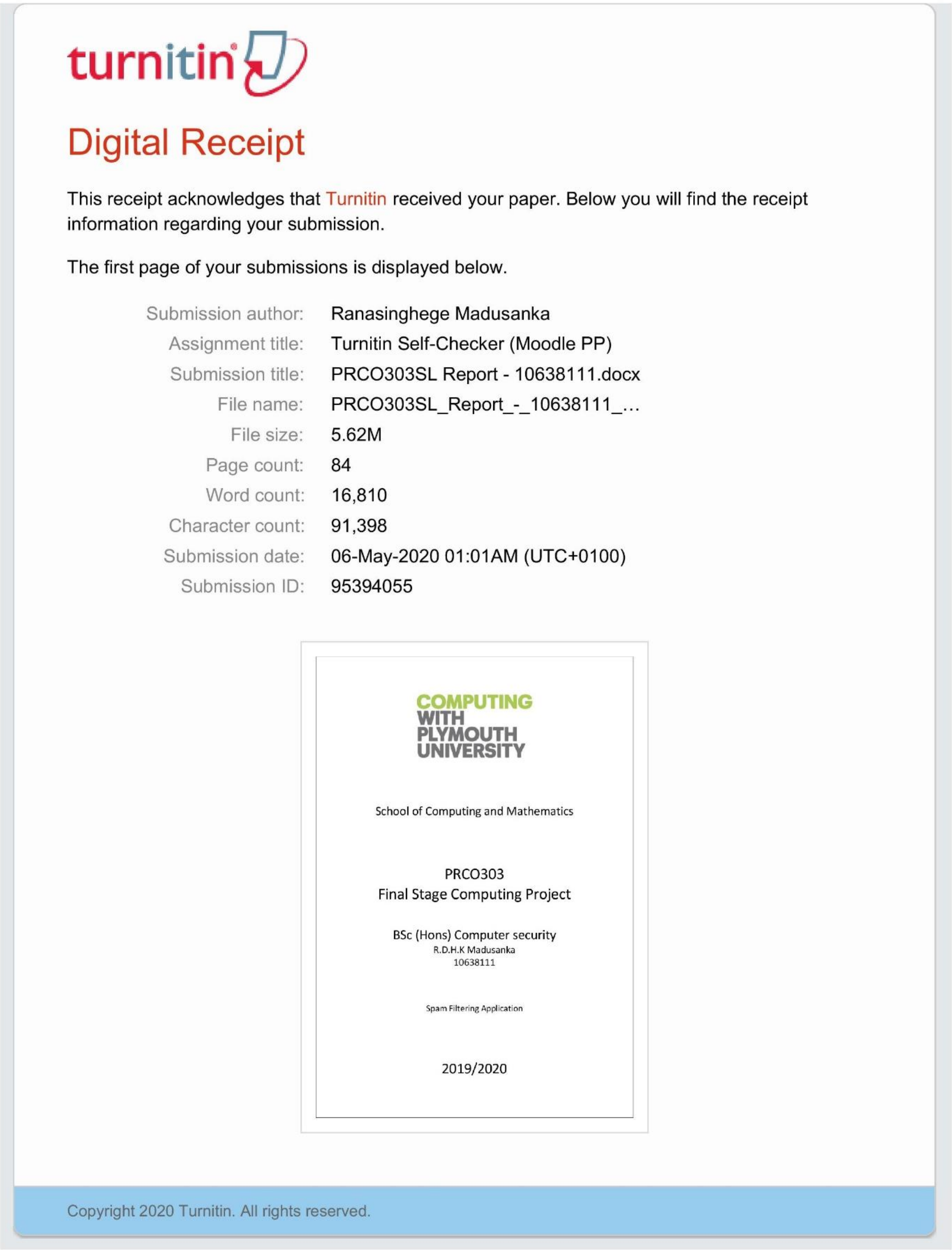
Items to be completed before the next supervisory meeting:

full Draft of the Background [illegible].
Draft of th Method & Approach

20/02/2020
Supervisor (Signature & Date)

Figure 38: Meeting minutes 4

19.8. Turnitin digital receipt



Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

Submission author:	Ranasinghege Madusanka
Assignment title:	Turnitin Self-Checker (Moodle PP)
Submission title:	PRCO303SL Report - 10638111.docx
File name:	PRCO303SL_Report_-_10638111_...
File size:	5.62M
Page count:	84
Word count:	16,810
Character count:	91,398
Submission date:	06-May-2020 01:01AM (UTC+0100)
Submission ID:	95394055

COMPUTING WITH PLYMOUTH UNIVERSITY

School of Computing and Mathematics

PRCO303
Final Stage Computing Project

BSc (Hons) Computer security
R.D.H.K Madusanka
10638111

Spam Filtering Application

2019/2020

Copyright 2020 Turnitin. All rights reserved.

Figure 39: Turnitin digital receipt

NOTES:

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

……………………………………………………………………………………………………

……„……………………………………………………………………………………………………