جامعة الملك سعود
King Saud University
كلية علوم الحاسب والمعلومات
قسم تقنية المعلومات

CYBERPHISH

CyberPhish

IT 497: Graduation Project Report
Product Release-2

Prepared by
Reema Alkraidees, 441203734
Reema Altayash, 441201295
Leen Alluhaidan, 441201196
Jood Alhamlan, 441200934

Supervised by
Dr. Alia Alabdulkarim
Dr. Nora Alhammad

Second Semester 1444
2022/2023

Table of Contents

List of Tables

List of Equations

List of Figures

CyberPhish

Reema Altayash [1], Reema Alkraidees [2], Leen Alluhaidan [3] and Jood Alhamlan [4]

[1]Information Technology Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia; 441201295 @student.ksu.ed.sa
[2]Information Technology Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia; 441203734@student.ksu.ed.sa
[3]Information Technology Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia; 441201196@student.ksu.ed.sa
[4]Information Technology Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia; 441200934
@student.ksu.ed.sa

Abstract (English): CyberPhish is a phishing email detection approach that utilizes a support vector machine classification model proposed and can be utilized as a part of a solution for anti-phishing. An email user can use this solution to distinguish phishing emails from the rest of the emails in the inbox. In this approach, SVM or Support Vector Machine is put to use. Evaluating the parameters like the word vectorizer, oversampling, and preprocessing methods, SVM finds the fittest decision boundary for categorization so that newly received email data can be correctly classified as phishing or legitimate. Preliminary experiments show that this approach is practical for detecting phishing emails with minimal false negatives at a speed adequate for mobile applications, and an accuracy of 97%. This classification is stored in the database and presented to the user via the interface when the application syncs the user's email inbox. The application provides an awareness section that features selected awareness articles, games, and a quiz.

Abstract (Arabic) : سايبرفيش هو تطبيق يساعد المستخدمين للكشف عن رسائل البريد الإلكتروني الاحتيالية التصيدية باستخدام خوارزمية آلة متجه الداعم.**(SVM)** يمكن لمستخدم البريد الإلكتروني استخدام هذا الحل لمكافحة التصيد الاحتيالي. في هذا النهج، يتم استخدام خوارزمية **SVM** تصنيف وتمييز رسائل البريد الإلكتروني الاحتيالية عن غير الاحتيالية. من خلال تقييم عدة عوامل مثل الكلمات وتأثيرها بالتصنيف الناتج، زيادة العينات المستخدمة، وطرق المعالجة المسبقة لتنقيح النصوص، تجد خوارزمية **SVM** حدود القرار الأنسب للتصنيف بحيث يمكن تصنيف بيانات البريد الإلكتروني المستلمة حديثًا بشكل صحيح على أنها احتيالية أو صالحة. تظهر التجارب الأولية أن هذا النهج عملي ويساعد على الكشف عن رسائل البريد الإلكتروني المخادعة بأقل قدر من معدلات الخطأ وبسرعة مناسبة لتطبيقات الهاتف المحمول. يتم تخزين التصنيف في قاعدة البيانات ويتم تقديمه للمستخدم عبر واجهة التطبيق، وذلك عندما يقوم التطبيق بمزامنة صندوق البريد الإلكتروني للمستخدم. يتم عرض النتائج في التطبيق لإعطاء نظرة شاملة فيما يتعلق بصندوق البريد الإلكتروني للمستخدم، على سبيل المثال، باستخدام مخطط دائري ومخطط خطي وإحصاءات ونسبة الخطر والتبرير لهذه النسبة والتصنيف. يوفر التطبيق قسمًا للتوعية يحتوي على مقالات توعوية وألعاب واختبار قصير.

Keywords: Cybersecurity; Phishing; Artificial Intelligence; Detection; Awareness

1 Introduction

Have you ever been tricked into sending your credit card information to an unknown source via email? If your answer is "yes", then unfortunately you got phished by an attacker. Typically, a simple reply to an email can cause great damage to you and your possessions.

According to the UK IT Governance [1], phishing is accomplished by sending messages that appear to be from a legitimate company or website. The message will typically include a link that directs the user to a bogus website that appears to be legitimate. After that, the user is prompted to enter personal information, such as their credit card number: this information is then used to steal the person's identity or to make fraudulent credit card charges.

Phishing has reached epidemic proportions and affects people regardless of their age. According to the 2018 Data Breach Investigations Report "Phishing and pretexting represent 98% of social incidents and 93% of breaches" [2]. The attacks have become so advanced and legitimate-like that neither the user nor email service provider can distinguish between phishing and real emails.

It was stated in the ThreatLabz 2022 phishing report that phishing attacks rose 29% in 2021 compared to 2020 [3]. Every quarter, Check Point tracks the brands that are most frequently imitated by hackers. Typically, Microsoft is at the top of this list, but in the fourth quarter of 2021, DHL replaced Microsoft as the most imitated brand in phishing attempts. In the third quarter, DHL accounted for 9% of all phishing emails, while in the fourth quarter, it skyrocketed up to 23%, Which highlights how users are unaware of phishing emails [4]. One of the key findings of the 2018 Data Breach Investigations Report [2], and the 2022 ThreatLabz report is that email continues to be the top phishing vector [3].

People while waiting for an unexpected shipment, may receive an email purporting to be from a shipping company but actually being sent by an attacker attempting to phish them. People's naivety and lack of awareness allow for these types of attacks to occur.

For Gmail users, this is where CyberPhish comes into play. CyberPhish is a phishing detection mobile application that gives Gmail users a feedback analysis and warns them regarding an email. Gmail users who depend on email on a day-to-day basis might find it tiresome to analyze every single email received to decide if it is safe or not. Similarly, Gmail users using email might wish to have an easier, faster, and better solution to figure out the

legitimacy of an email sent to them. CyberPhish will be like a safety net that opens Gmail users' eyes to the true purpose of the emails in their inbox.

Our proposed solution is to create a mobile application that makes use of the artificial intelligence algorithm SVM, since it has great results when it comes to supervised learning in our domain, according to Java Point [5]. The algorithm is used to analyze email data, specifically Gmail, to classify whether the email is a phishing attempt or not. The dataset will be used to train the model. Considering the non-technical users that can benefit from our application, the CyberPhish application will include educational materials that enlighten users with the necessary knowledge of phishing, including its definition, techniques, background, and other details. The CyberPhish application will be maintaining the security principle of integrity, and it aims to limit the possibility of users falling for phishing attacks through emails by warning them beforehand and foreseeing if the email was phishing.

Users who are using Gmail might face some issues when it comes to unknown emails, like if this email is legitimate or not. CyberPhish will resolve this issue since it offers users a feedback analysis regarding their email. CyberPhish will achieve its aim of reducing the likelihood of consumers falling victim to phishing attempts by implementing the features listed.

- Allow the user to login to Gmail.
- Authorize our application to sync and read the user's email.
- Extract the user's email data.
- Analyze email data using AI.
- Provide feedback about the email to the user.
- Provide awareness content articles to increase user's knowledge in phishing attacks.
- Provide statistical analysis reports of the user's inbox activity in the form of graphs within a specified time period.

CyberPhish will be designed for Gmail users who are unsure if an email they have received is legitimate. The application will analyze email data using artificial intelligence and security principles to determine whether the email is phishing or not. The application is in the English language and supports Android mobile applications.

To develop such an application, will be coded using front-end languages and frameworks as well as back-end languages through various programming IDEs. An AI algorithm will be used in the process of developing this application to determine whether an

email is a phishing attempt or not by analyzing the email's data. The SVM algorithm will be trained using the dataset. The application will provide activities and trivia to help raise user awareness towards email phishing. In addition to producing visual analysis on user demand by means of charts during a specified time interval. However, CyberPhish does not send emails, replies, or forwards, nor does it block the phishing attacks; it will warn users of detected phishing emails with reason and probability. Finally, the application will be available in English for Gmail users.

For Gmail users who are unaware of phishing emails, CyberPhish is a mobile application that detects phishing emails and warns the user, unlike Chrome's Email Phishing Tool extension, our product is a mobile application that makes it portable for users to check their emails wherever they are.

The implementation of CyberPhish began with conducting interviews and distributing surveys to email users, followed by data analysis to understand and evaluate the users' needs. According to the user's requirements, an application was created that automatically analyzes the emails in the user's inbox and provides straightforward feedback analysis for users of various technical backgrounds. After collecting phishing and non-phishing emails from online sources, the team's own emails, and the emails of certain acquaintances, some extra preprocessing was performed on the obtained emails by Python, rows that were duplicated or missing were eliminated, leaving just the original row. To balance the two datasets, the Date and Content type columns were also removed, since the phishing dataset did not have those attributes. There were no outliers since each row is unique, which means no deviations from other values. Encoding was unnecessary for the dataset because it contained just unique emails with varied messages, which is the goal of the AI method. After that, the model was then trained by the given dataset in order to classify phishing and non-phishing emails. In the context of determining whether an email is a phishing attempt or legitimate, the process involves several stages, including classification by a model, analysis of sender and link reputation, and calculation of a percentage. A variety of parameters used in the calculation including the email body language, sender reputation analysis, and link reputation analysis if a link was detected in the email body. If the email is classified as legitimate, no percentage is displayed. However, if the email is classified as phishing, a percentage is displayed, along with a risk label indicating the severity of the threat. The calculation of the percentage is based on various countermeasures and techniques designed to prevent phishing attacks, such as those defined by the Open Web Application Security Project (OWASP). After completing

the application, the team have performed user acceptance testing on 20 Gmail users with varying technical experience. The objective was to assess the performance and efficiency of the overall application. The users who took part in the testing were enthusiastic about CyberPhish and recognized its significance.

This project contributes significantly to the areas of technology and study. For starters, it provides a public dataset with over 1105 phishing emails and 3637 legitimate. The dataset was built manually by the team with a combination of online resources and the team's personal emails.

Secondly, the project offers an automated solution that is free of charge and easy to use. By utilizing the features that are available for mobile applications, CyberPhish can easily notify the user anytime and anywhere as opposed to a desktop application, website, or browser extension. Due to the fact that CyberPhish is aimed for both individuals and companies alike, it can greatly reduce the potential of successful phishing attacks on personal and organization devices.

The risks of dangerous email communications are increasing as the Kingdom of Saudi Arabia moves toward digital transformation. CyberPhish will help Saudis distinguish which email messages are genuine and which are fake. Furthermore, CyberPhish will assist users in better understanding and protection against threats that endanger the security of their regular communications.

Email exchanges have become increasingly frequent in our busy lives, yet many individuals pay less attention to the minor details in their emails. These specific details are frequently a clear indicator of a phishing attempt. CyberPhish tackles this problem by considering the advancement of artificial intelligence technologies and employing them to design the solution, where we pay attention to the little details so that the users do not need to. As the world debates online safety methods, they must consider the communication safety of those who use this cyberspace.

Unlike previous solutions, CyberPhish offers features for detecting phishing emails, notifying the user, giving feedback analysis, providing statistical reports in the form of graphs, and providing phishing awareness content on mobile phones, free of charge, and aimed at individuals and organizations.

In this report, we will begin with a background section that covers the needed domain knowledge, followed by a literature review that includes a competitive product analysis.

Following that, the system design and development section includes our methodology, system requirements, which contain system users, requirement elicitation and analysis, and user interactions, as well as a road map and our user stories in the product backlog for both functional and non-functional requirements. The architecture diagram, class diagram, and component level design will then be shown in the system design section. Then we will show off our data design, which comprises data models, data collecting and preparation, interface design and implementation, and system evaluation, which includes experimental results, user acceptance testing, participant demographics, and questionnaire responses. In addition, we will use NFR testing to measure the quality attribute and then analyze the results. Then followed the conclusion and future work, as well as acknowledgements for those who contributed to the success of CyberPhish, Finally, we will include references as well as appendices for the interviews, questionnaires.

2 Background

To know more about CyberPhish, first, we would explain the context of our domain, cyberattacks in general, and particularly phishing, as well as key concepts and terms, proposed artificial intelligence algorithms, and the external software that will be used to help us deliver CyberPhish.

2.1 Cyberattacks

In the last year, cyber criminals delivered a wave of cyber-attacks that were not just highly coordinated but far more advanced than ever before seen [6]. Any aggressive action against computer information systems, computer networks, infrastructures, or personal computing devices is referred to as a cyberattack. in an attempt to profit from damage or eliminate a specific target. Cyberattacks can take many different forms, from the placement of malware on a user's computer to attempting to compromise the infrastructure of whole countries. Malware, phishing, ransomware, man-in-the-middle attacks, and other tactics are used by cybercriminals to initiate cyberattacks.

Our project domain is cyber-attacks. We were able to gather information about our domain from online resources such as Google Scholar, existing systems such as Mime Cast, and articles written by domain experts on ResearchGate, to learn more about email protection terms and concepts, and to provide the user with proper detection and feedback about phishing emails.

2.2 Phishing

Proofpoint defined phishing attacks as “when attackers send malicious emails designed to trick people into falling for a scam. Typically, the intent is to urge users to reveal financial information, system credentials, or other sensitive data” [7]. Topped the list as one of the most active threats in Cisco's 2021 Threat Report [6]. Since it's less about technology and more about social engineering, phishers use manipulation techniques that exploit human error. These "human hacking" techniques are commonly used in cybercrime to trick unwary users into disclosing data, dispersing malware infections, or granting access to restricted systems [8].

Even though professional email services now offer some degree of security against phishing emails, they are by no means faultless. Therefore, focusing on detecting phishing emails should be considered a different aspect of protecting emails.

2.3 AI in phishing detection

Artificial intelligence methods can be used to improve the detection of phishing emails [9] and these methods are the focus of our solution. Massive volumes of data can be processed by artificial intelligence far more quickly than by human brains. The AI techniques that we will implement in our phishing detection software are SVM, Naïve Bayes, and Random Forest.

2.3.1 Support Vector Machine (SVM)

SVM is a well-known supervised learning algorithm for classification and regression problems. However, it is primarily used in machine learning for classification problems. The SVM algorithm's goal is to find the best line or decision boundary for categorizing n-dimensional space so that we can easily place new data points in the correct category in the future [5].

Each data item is plotted as a point in n-dimensional space, where n is the number of features you have, with the value of each feature being the value of a certain coordinate. After that, we execute the classification by locating the hyperplane that best distinguishes the two classes as shown in Figure 1.

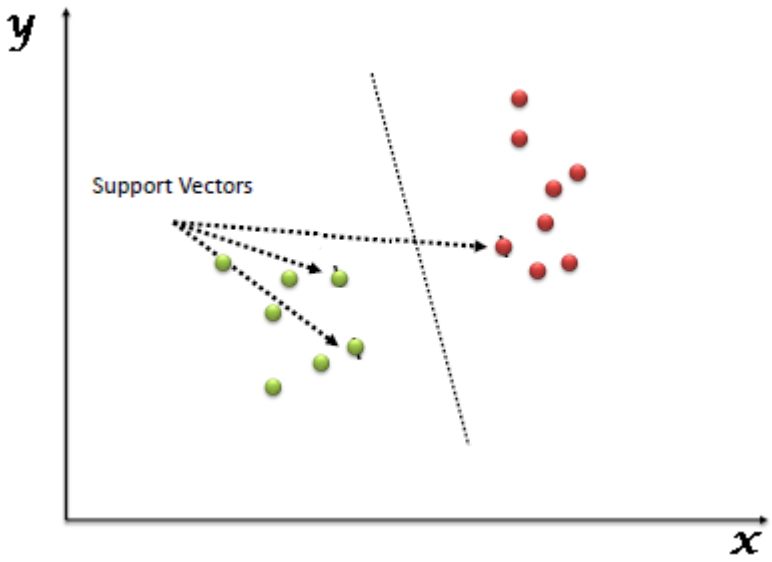


Figure 1:SVM n-dimensional space in 2.3.1

Support Vectors are simply the coordinates of individual observation. The SVM classifier is a frontier that best segregates the two classes (hyper-plane/ line) [10].

SVM techniques rely on the kernel, which is a collection of mathematical functions. The role of the kernel is to transform attribute space into a new feature space that fits the

maximum-margin hyperplane, allowing the method to generate the model with linear, polynomial, RBF, and sigmoid kernels [11]. Hence, in general, the Kernel Function modifies the training set of data so that a non-linear decision surface can transform to a linear equation in a larger number of dimension spaces.

2.3.2 Naïve Bayes

The Naïve Bayes algorithm is a supervised learning algorithm for classification problems. It is primarily used in text classification, which includes a large training dataset. It is one of the simplest and most effective classification algorithms, and it aids in the development of fast machine learning models capable of making quick predictions. This algorithm uses the Bayes' theorem to determine the probability of a hypothesis with prior knowledge as shown in Equation 1.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Equation 1: Bayes Theorem in 2.3.2

It is a probabilistic classifier, which means it predicts based on the probability of an object. The Naïve Bayes algorithm is used for spam filtration and sentiment analysis [5].

2.3.3 Random Forest

Random Forest is a well-known machine learning algorithm from the supervised learning technique. It can be applied to both classification and regression problems in machine learning. It is based on the concept of ensemble learning, which is a process that involves combining multiple classifiers to solve a complex problem and improve the model's performance [5].

It is a classifier that contains several decision trees on various subsets of a given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on a single decision tree, the random forest takes the predictions from each tree and predicts the final output based on the majority vote of predictions. This algorithm takes less training time, predicts output with high accuracy, and maintain accuracy when a large proportion of data is missing. The greater the number of trees in the forest, the greater the accuracy and the lower the risk of overfitting [5].

2.3.4 Performance Measures

Classification is a type of supervised machine learning problem where the goal is to predict the class for one or more observations. In CyberPhish, four performance matrices will be used: accuracy, recall, precision, and the F1 score, in addition to the confusion matrix. These performance indicators were chosen based on a research paper by Abdul Basit, Maham Zafar, Abdul Rehman Javed, and Zunera Jalil, that worked on the same domain as CyberPhish [14].

A **confusion matrix** is an extremely useful tool to observe in which way the model is wrong. It is a matrix that compares the number of predictions for each class that are correct and those that are incorrect as seen in Figure 2.

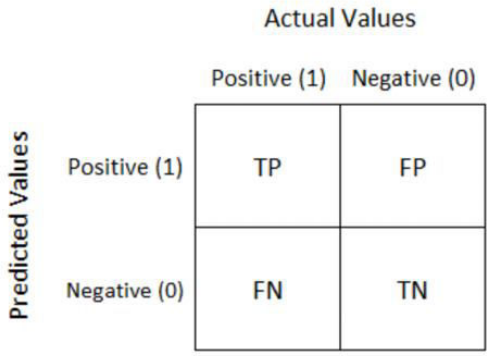


Figure 2: Confusion matrix in 2.3.4

In the equations of accuracy, recall, and precision the true positive (TP) means the number of correct classifications of the positive examples. On the other hand, false negative (FN) means the number of incorrect classifications of positive examples. The false positive (FP) is the number of incorrect classifications of negative examples. And the true negative (TN) is the number of correct classifications of negative examples.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Equation 2: Accuracy formula in 2.3.4

$$Recall = \frac{TP}{TP + FN}$$

Equation 3: Recall formula in 2.3.4

$$Precision = \frac{TP}{TP + FP}$$

Equation 4: Precision formula in 2.3.4

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Equation 5: F1 Score formula in 2.3.4

Accuracy calculates the number of times a model made correct predictions across the entire dataset [12]. It is the number of correct predictions divided by the total number of predictions as seen in Equation 2. The model's recall determines how many of the positive class samples in the dataset were properly identified [12]. Recall is the number of correctly classified positive examples divided by the total number of actual positive examples as seen in Equation 3 [13]. Precision is a measure of how many of the model's "positive" predictions were right [12]. Also, precision refers to the number of correctly classified positive examples divided by the total number of examples that are classified as positive as seen in Equation 4 [13]. Lastly, The F1 score is a machine learning evaluation metric that measures a model's accuracy. It sums up the predictive performance of a model by combining precision and recall as seen in Equation 5.

2.3.5 Cross Validation

Cross-validation is a statistical method for estimating machine learning model skill. It is often used in applied machine learning to compare and pick a model for a specific predictive modeling problem since it is simple to grasp, simple to implement, and produces skill estimates with lower bias than other methods [15].

- K-Fold Cross-Validation

Cross-validation is a resampling approach that is used to evaluate machine learning models using a small set of data. The process has a single parameter called k that defines the number of groups into which a given data sample should be divided. As a result, the process is frequently referred to as "k-fold cross-validation." When a specific value for k is chosen, it can be used in place of k in the model's reference.

In applied machine learning, cross-validation is generally used to evaluate the performance of a machine learning model on new and unseen data. A small sample size will

be used to test how the model will perform in general when used to make predictions on data that was not utilized during the model's training.

It is straightforward to grasp since it produces a less biased or optimistic estimate of the model's performance than other methods, such as a simple train-test split.

The general procedure is as follows:

1. Shuffle the dataset randomly.
2. Divide the dataset into k groups.
3. For each unique k group:
   3.1. Take the group as a holdout or test data set.
   3.2. Take the remaining groups as a training data set.
   3.3. Fit a model on the training set and evaluate it on the test set.
   3.4. Retain the evaluation score and discard the model.
4. Summarize the skill of the model using the sample of model evaluation scores.

Importantly, each observation in the data sample is assigned to an individual group and stays in that group for the duration of the procedure. This means that each sample is used in the holdout set once and then used to train the model K-1 times [15].

- Leave-One-Out Cross-Validation

The LOOCV or leave-one-out cross-validation approach is used to measure the performance of machine learning algorithms when they are used to generate predictions on data that was not used to train the model [16].

It is a computationally expensive approach that yields an accurate and unbiased evaluation of model performance. Although the process is simple to use and requires no configuration, there are situations when it should not be utilized, such as when evaluating a large dataset or a computationally expensive model [16].

LOOCV, is a variant of k-fold cross-validation in which k is equal to the number of samples in the dataset. LOOCV is the most computationally expensive version of k-fold cross-validation. For each sample in the training dataset, one model must be generated and assessed [16].

The benefit of having so many fit and evaluated models is a more robust assessment of model performance since each row of data is given the opportunity to represent the whole test dataset [16].

Because of the computational expense, LOOCV is not suitable for very large datasets with tens or hundreds of thousands of samples, or for models that are expensive to fit, such as neural networks [16]. Another issue with LOOCV is that it is prone to high variance or overfitting because we feed the model practically all of the training data to learn and only a single observation to assess [16]. Also, since the dataset used for CyberPhish has no relations between the tuples, there is no reason or benefit from applying LOOCV.

2.4 External Software

To assist us on our journey, we will use the Gmail API, Kommunicate, DialogFlow, Orange data mining, and APIVoid as the external software, which will be used to interact with users' Gmail inboxes via our application. The Gmail API can be used to access users' Gmail mailboxes and is suitable for read-only mail extraction [17]. The Kommunicate and DialogFlow will be used on the chatbot to specify the intent type and response type, respectively, so that the user can interact with the chatbot effectively [20][21]. Orange is an open-source data visualization, machine learning and data mining toolkit. Orange is used to train and test the three AI techniques [18]. APIVoid helps with automating the manual work of security analysists by offering JSON APIs for cyber threat analysis, detection, and prevention [19].

3 Literature Review

The use of artificial intelligence models in phishing detection has recently been the subject of numerous studies, investigations, and products. In this chapter, some of the developed studies and competing software will be discussed.

3.1 Competitive Product Analysis

CyberPhish has many big-name competitors. We investigated the features of each competing software offered as well as those lacking in order to find the best features for our users.

3.1.1 Avanan

Figure 3: Avanan's Logo in 3.1.1

The first email security solution to be made available as an app in Office 365 and Google Workspace was Avanan. Its patented technology deploys after the default Microsoft and Google filters but before the inbox, So, blocking attacks before they reach the inbox. A subscription charge based on the business is utilized to protect the whole workplace collaboration suite with Avanan [22].

3.1.2 Cofense PDR

Figure 4: Cofense's Logo in 3.1.2

Cofense introduces the industry's first cloud-native email security for Microsoft 365 and Google Workspace, deployable in under a minute. The Phishing Detection and Response (PDR) platform leverages a global network of nearly 32 million people actively reporting

suspected phishing, combined with advanced automation. Cofense serves over 2,000 enterprise customers [23].

3.1.3 Mimecast

Figure 5: Mimecast's Logo in 3.1.3

Mimecast is one of the biggest competitors to CyberPhish. They mainly target businesses rather than individual users. They provide phishing detection as a proactive threat detection, which means they take action against these malicious emails. They also offer feedback analysis. They support many email clients, such as Outlook, Google Workspace, and other business email clients. Mimecast supports mobile applications, despite the fact that they are not intended for personal use and do not provide free service [24].

Figure 6: PhishTector's Logo in 3.1.4

3.1.4 PhishTector-Chrome's Extension

This phishing detection tool is a plug-in Chrome extension that reads the email, analyzes it, and provides feedback, although based on our experience, we did not receive any type of feedback. It does not provide a Gmail API or any other email client. The user supposed to log in into his account, select an email, then the tool will analyze the email and provide feedback. Furthermore, it targets individual users and offers free usage. PhishTector does not support mobile versions and does not block attacks automatically [25].

3.1.5 Email Veritas email add-on

Figure 7: Email Veritas Logo in 3.1.5

Email Veritas involves a light browser add-on and provides an interface for users to configure and customize the Email Veritas anti-phishing service delivered to them. The software protects business email against phishing attacks by personalizing the protection. EmailVeritas is made for individual user messaging habits. Phishing messages are flagged and differentiated from legitimate ones using a color-coding scheme, e.g., red (high risk), orange (medium risk), green (safe) and warning (caution). That way, the recipient can decide which emails to read and which ones require a greater level of caution. When EmailVeritas is deployed, it automatically provides the user a free 30-day trial for up to 10 users. After the trial period, the user must purchase a yearly subscription-based license and a minimum of 10 users per order [26][27].

1.1 Competitors' summary table

Table 1: Competitors' Summary in 3.2

Feature	**Avanan**	**Cofense PDR**	**Mimecast**	**PhishTector-Chrome's Extension**	**Email Veritas email add-on**	**CyberPhish (proposed)**
Mobile application	✗	✗	✓	✗	✗	✓
Gmail sign in	✗	✗	✗	✓	✓	✓
Provide feedback	✓	✓	✓	✗	✓	✓
Intended for individuals	✗	✗	✗	✓	✓	✓
Free usage	✗	✗	✗	✓	✗	✓
Allows Microsoft products sign in	✓	✓	✓	✗	✓	✗
Block attacks automatically	✓	✓	✓	✗	✗	✗

4 System Design and Development

4.1 Methodology

The application "CyberPhish" was developed using an agile technique. The agile approach is defined as an iterative method to software development and project management that aids in the delivery of valuable increments of the product. The agile methodology was implemented by going through 5 sprints, reevaluating and amending our plans on a regular basis to stay on schedule and adapting to changing requirements, and lastly releasing two releases.

There are three roles in the scrum framework: product owner, developer, and scrum master. "CyberPhish" had a single product owner who established the vision and set the priorities for "CyberPhish." In addition, the product owner served as a connection between the stakeholders and the developers. The developers have the abilities required to fulfill the product vision, create deliverable features, breakdown user stories, react to changes, and bring the vision to life. Finally, there is the scrum master, who coaches and advises the team to ensure that the ideals and principles of agile methodology are followed, arranges meetings, and gives feedback.

During the "CyberPhish" application life cycle, five events took place, starting with the sprint that set the sprint's timeframe. Then, at the start of each sprint, we determine which user stories will be worked on in this sprint based on their priority in the product backlog, as well as how they will be developed and completed.

There is a daily scrum meeting during the sprint to assess progress and how near we are to the sprint objective, examine problems that emerged during the sprint, and refigure how to address those problems. Furthermore, a sprint review was held between "CyberPhish" developers and the product owner in order to gather input, evaluate problems, and respond to changes. Prior to the start of the next sprint, a sprint retrospective meeting is held to review the process, the team's progress, and to highlight any issues that have arisen during the sprint so that they can be averted in the upcoming sprint.

There are three scrum artifacts that “CyberPhish” is expected to deliver:

- Product backlog: it displays a list of features ordered by priority as user stories with acceptance criteria, that are expected to be developed throughout sprints.

- Product increment: At the end of each sprint, "CyberPhish" produces a product increment in which we merge the previous increments with the current sprint increment.
- Sprint backlog: a list consisting of the chosen user stories from “CyberPhish” product backlog to be work on and developed during the sprint.

The tools that have been used to achieve the “CyberPhish” objectives are GitHub [1]and Jira[2]. Jira is software that assists developers in tracking bugs and issues, organizing tasks, and manage agile teams. This software has simplified the project management process by setting the system requirements, tracking the progress, and documenting what was achieved as meeting notes. Meanwhile, GitHub is a website and cloud-based service that helps developers store and manage their code, as well as track and control changes to their code [28]. We were able to easily handle the collaboration of "CyberPhish" application and machine learning model codes through GitHub, commit new modifications among the "CyberPhish" team, and observe these modifications on the source code with no conflicts or missing components.

4.2 System Requirements

In this chapter, we will cover system design. It is a creative activity in which we identify the software components of CyberPhish as well as the relationships based on the system requirements of the CyberPhish user.

4.2.1 System Users

Potential users of our system are Gmail users who will use the application to detect phishing emails, have at least an intermediate education level, and are familiar with the English language. CyberPhish needs little technical experience to do things such as signing in with Gmail, mailing.

4.2.2 Requirements Elicitation and Analysis

Stakeholders in CyberPhish played a major role and were kept in mind during our requirements elicitation process. These stakeholders included Gmail users.

[1] GitHub: https://github.com/Cyberphish/2022-GP1-G8
[2] Jira: https://2022-1st-gp8.atlassian.net/jira/software/projects/GP8/boards/1

To learn more about our users, we distributed a questionnaire with six closed-ended questions and one open-ended question about what potential users thought about various topics. The questionnaire asked if users were familiar with the term "phishing," how frequently they receive phishing emails, where they receive the most phishing emails, if a phishing detection mobile application is more accessible and easier to use, if they will authorize CyberPhish to access their emails, the method of feedback they prefer, and suggestions the CyberPhish team should consider.

We received about 400 responses, and according to the questionnaire results, we discovered that 54% of users were familiar with the term “phishing”, yet 46% of the responses were divided between “no”, and “maybe”. 80.5% of users responded that they receive phishing emails often. 65.5% of the users agreed that their personal email is where they receive the majority of the phishing emails. 90% of CyberPhish’s potential users agree that a phishing detection mobile application will ease their use and increase accessibility. 68.5% of the respondents agreed to allow CyberPhish to be authorized to access their email, and the ones that disagreed mentioned in the open-ended question that they would authorize CyberPhish if there was a clear privacy policy. 85% of the users agreed that the best way to receive feedback is in a brief description, percentages, or list of indicators kind of way. For more details, see Appendix A.

In the open-ended question, we received amazing suggestions that can help us now and for future work. The responses to this question gave us a look at what our potential users expect of us. Some of the superb replays we got were:

- Include feedback on best practices to counter these frauds.
- Add a confidentiality agreement that CyberPhish won’t maliciously use or leak the user’s emails.
- Give the users the option to authorize CyberPhish with regard to saving their phishing emails or not for training purposes.
- Automatically block high-probability phishing emails.
- Make the mobile application available for free.
- Raise the user’s awareness regarding phishing.

4.2.3 User Interactions

Figure 8 illustrates the system's function and CyberPhish's user starts with logging in using their Gmail account via Gmail API, then analyzed their emails by CyberPhish's model and APIVoid. Furthermore, the user can view their inbox, read emails and its details, filter it depending on the email flag. The analysis feedback is viewed as a flag by default for all the categories, the percentage and reasons feedback are presented for the phishy emails only. The reasons represent the risk scores of the link, the sender and the email langueage, where the user has the choice to view the triggering phishy words. Moreover, the analytics report shows the analysis of the received emails using a linear chart, insights, and a pie chart, after choosing the preferred time frame, where the 'yearly' choice is chosen by default. The user may raise their awareness by reading the awareness content and the articles, and test their awareness by taking the quizz. CyberPhish answers the user's questions via the chatbot. Lastly, the user can log out as the end of their journey in CyberPhish application.

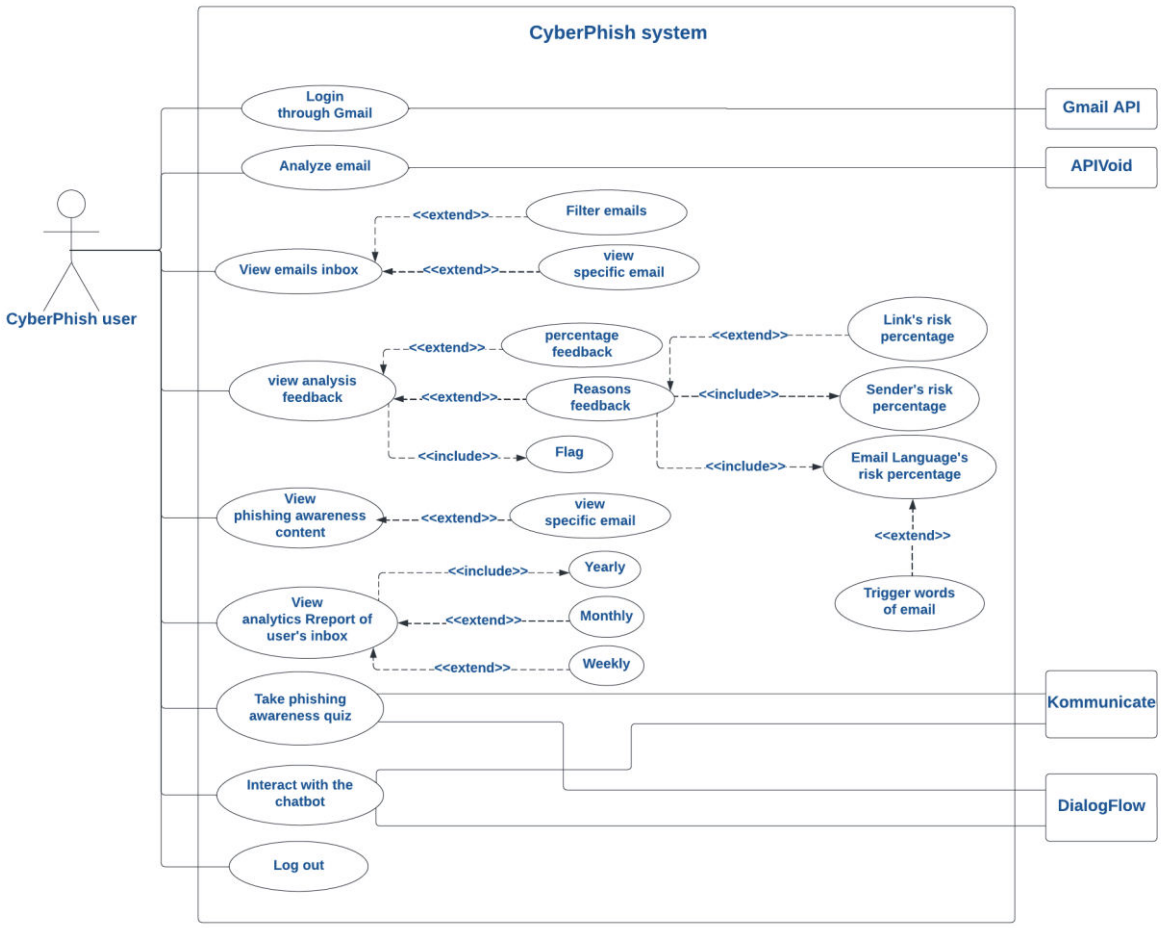


Figure 8: User Interaction in 4.2.3

4.2.4 Roadmap and Product Backlog

- Roadmap

As seen in Figure 9 CyberPhish's implementation is divided into five sprints, with a number of tasks to be completed in each sprint.

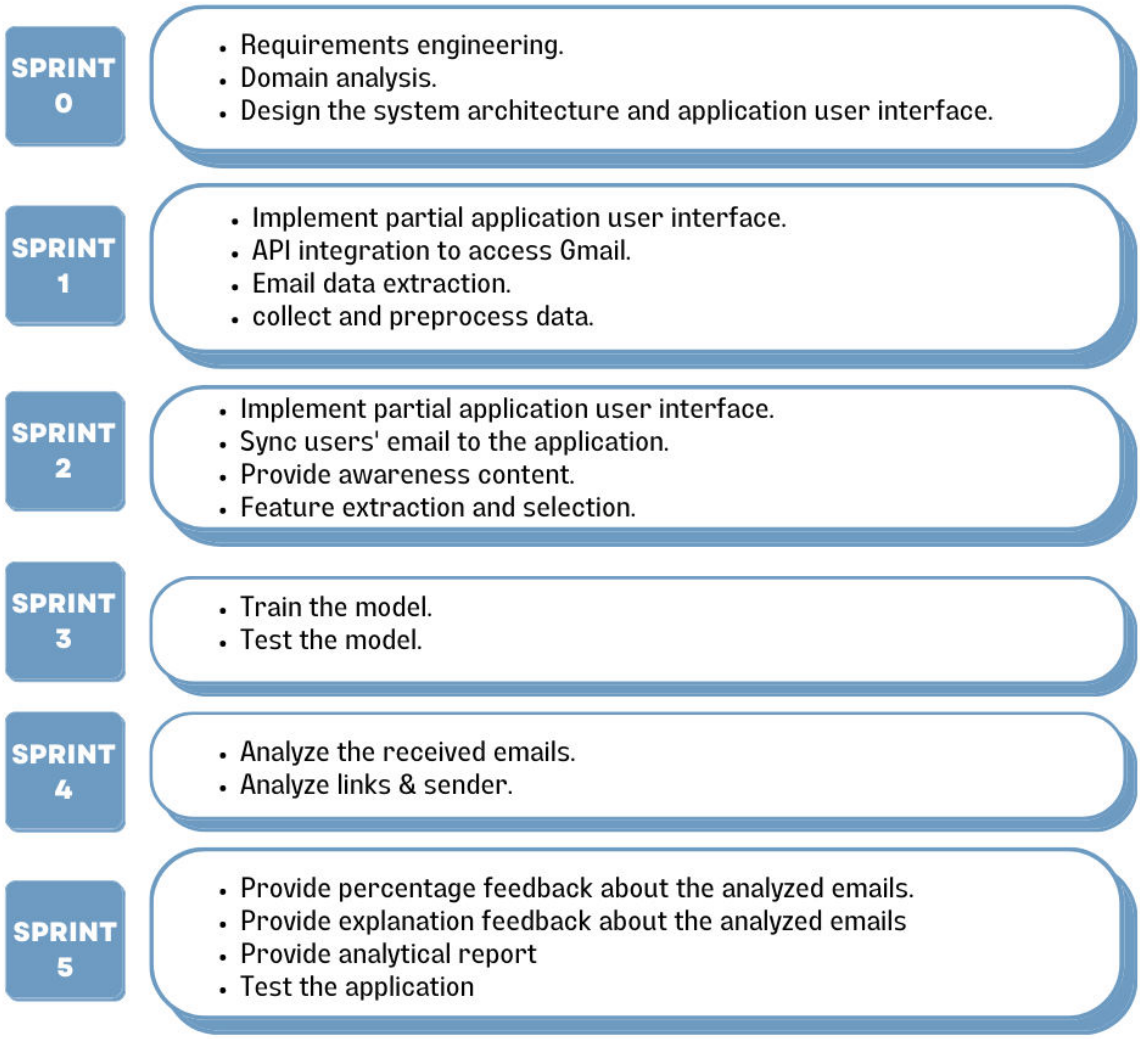


Figure 9: Roadmap in 4.2.4

The product backlog refers to a prioritized list of functionalities that products should contain. The acceptance criteria for each user story were first written. Then, we established our definition of ready, a standard that user stories must meet before they can be added to the sprint.

Table 2: Definition of Ready in 4.4

Definition of Ready
☐ Acceptance criteria are clear and testable

- ☐ Details of user stories are sufficiently understood
- ☐ Story estimated and small enough to be completed during sprint
- ☐ Functional tests passed.
- ☐ Non-functional requirements met.

- Product Backlog Table

Table 3: Product Backlog in 4.4.1

ID	PBIs (User Stories)	Size	Type (Feature, defect, technical work, knowledge acquisition)	Status (To do, in progress, or done)	Acceptance Criteria The conditions of satisfaction that must be met for that item to be accepted.
GP8-1	As a CyberPhish user, I want to sign-in to my Gmail account, so that I can access the application and use it.	2	Feature	Done	1. As a CyberPhish user, if I click the sign in button, then I should be redirected to Google's sign in page to fill out the sign-in form with my email and password. 2. As a CyberPhish user, if I fill out Google's sign-in form with information, then the API should validate my information. 3. As a CyberPhish user, if my information was correct, then the API should redirect me back to the application to access my profile. 4. As a CyberPhish user, if my information was incorrect, then I should receive an error message.
GP8-2	As a CyberPhish user, I want to view my email inbox, so that I can read my emails.	1	Feature	Done	1. As a CyberPhish user, If I was properly signed in, then I should be able to view my email inbox.
GP8-3	As a CyberPhish user, I want my latest received emails to be analyzed, so that the application can detect the phishing emails.	3	Feature	Done	1. As a CyberPhish user, If I receive a new email, then it should be analyzed.
GP8-4	As a CyberPhish user, I want to know the probability of an email to be phishing in percentage, so that I can measure the risk of a received email.	3	Feature	Done	1. As a CyberPhish user, If I receive analysis feedback, then I should see a percentage of how likely the email is a phishing email.

GP8-5	As a CyberPhish user, I want to know the reasons why a certain email was flagged as phishing, so that I can know the different signs phishers usually use.	3	Feature	Done	1. As a CyberPhish user, If the analysis feedback is displayed in the interface, then I should be able to view the reasons. 2. As a CyberPhish user, if I decide to view why an email was flagged as phishy, then I can see three icons representing the risk percentage of the sender, links and language. 3. As a CyberPhish user, if I click the language icon, then I should be able to know the words that triggered the email flagging as a phishing attempt.
GP8-6	As a CyberPhish user, I want to view a statistical analysis about previously received emails, so that I can track my email status.	5	Feature	Done	1. As a CyberPhish user, If I enter my profile, then I should find a report page. 2. As a CyberPhish user, If I click on report page, then I should view a statical analysis about my previous received emails.
GP8-7	As a CyberPhish user, I want to view awareness content about phishing, so that I can be more aware and updated about it to avoid being a potential victim.	2	Feature	Done	1. As a CyberPhish user, if I enter the application, then I should be able to see a button that directs me to the awareness content. 2. As a CyberPhish user, if I clicked on the awareness content button, then I should be directed to the awareness content. 3. As a CyberPhish user, if I get directed to the awareness content, then I should be able to explore and read different topics about phishing.
GP8-8	As a CyberPhish user, I want to be able to have my questions about the app	2	Feature	Done	1. As a CyberPhish user, if I enter the application, then I should be able to see a button that directs me to

	answered through the chatbot, so that I can resolve any issue or confusion I faced while using the app.				the chatbot section. 2. As a CyberPhish user, if I tap on the chatbot button, then I should be directed to the chatbot conversation section. 3. As a CyberPhish user, if I get directed to the chatbot conversation section, then I should be able to start a conversation. 4. As a CyberPhish user, if I start a conversation with the chatbot, then I should be able to see an options button that displays frequently asked question. 5. As a CyberPhish user, if I tap on one of the questions, then I should be able to get an answer from the chatbot.
GP8-9	As a CyberPhish user, I want to be able to test my knowledge of phishing through quizzes in the chatbot, so that I can know my level of knowledge and what I need to learn more about	2	Feature	Done	1. As a CyberPhish user, if I enter the application, then I should be able to see a button that directs me to the chatbot section. 2. As a CyberPhish user, if I tap on the chatbot button, then I should be directed to the chatbot conversation section. 3. As a CyberPhish user, if I get directed to the chatbot conversation section, then I should be able to start a conversation. 4. As a CyberPhish user, if I start a conversation with the chatbot, then I should be able to see a quiz button that displays the quiz questions. 5. As a CyberPhish user, if I tap on one of the answer choices, then I should be able to get a reply from the chatbot that tells me if my answer was correct or not and then displays the next question.
GP8-10	As a CyberPhish user, I want to sign-out from my Gmail account, so that I can keep my email account secure.	1	Feature	Done	1. As a CyberPhish user, If I select the sign-out button, then I should be signed out from my account.

GP8-11	As a team member, I want to collect the dataset of both legitimate and fraudulent emails, so that it can be used in developing the classification model.	2	knowledge acquisition	Done	1. The dataset is collected for both legitimate and phishing emails to be used in the classification model.
GP8-12	As a team member, I want to preprocess the dataset, so that I have clean data to be used for modeling.	5	knowledge acquisition	Done	1. The dataset is cleaned so it can be usable for developing the model
GP8-13	As a team member, I want to extract appropriate features, so that I can train the model based on these features.	3	knowledge acquisition	Done	1. The features have been extracted and selected.
GP8-14	As a team member, I want to split the data, so that I can train and test the model.	3	knowledge acquisition	Done	1. The dataset is split into training and testing sets.
GP8-15	As a team member, I want to train the Naïve Bayes model using an acquired training dataset, so that I can build the model.	3	knowledge acquisition	Done	1. The Naïve Bayes algorithm is trained using the training dataset.
GP8-16	As a team member, I want to train the Random Forest model using an acquired training dataset, so that I can build the model.	3	knowledge acquisition	Done	1. The Random Forest algorithm is trained using the training dataset.
GP8-17	As a team member, I want to train the SVM model using an acquired training dataset, so that I can build the model.	3	knowledge acquisition	Done	1. The SVM algorithm is trained using the training dataset.
GP8-18	As a team member, I want to test the Naïve Bayes model using an acquired testing dataset, so that I can measure its performance.	3	knowledge acquisition	Done	1. The Naïve Bayes algorithm is tested using the testing dataset.
GP8-19	As a team member, I want to test the Random Forest model using an acquired testing dataset, so that I can measure its performance.	3	knowledge acquisition	Done	1. The Random Forest algorithm is tested using the testing dataset.
GP8-20	As a team member, I want to test the SVM model using an acquired testing dataset, so that I can measure its performance.	3	knowledge acquisition	Done	1. The SVM algorithm is tested using the testing dataset.

GP8-21	As a team member, I want to compare the testing results of the Naive base, Random Forest, SVM algorithms, so that I can choose the suitable algorithm for CyberPhish.	2	knowledge acquisition	Done	1. The test results show a clear indication of which algorithm is suitable for CyberPhish. 2. The test results show a clear indication of which algorithm is suitable for CyberPhish.
GP8-22	As a CyberPhish user, I want my emails to be synced, so that I have real-time updates.	5	Feature	Done	1. As a CyberPhish user, If I was properly signed in, then I should be able to receive new emails. 2. As a CyberPhish user, If I received a new email when I'm signed in, then it should be directly displayed in my email inbox.
GP8-23	As a CyberPhish user, I want to receive phishing email notifications, so that it keeps me aware.	3	Feature	Done	1. As a CyberPhish user, If I was properly signed in, then I should be able to receive phishing email notifications. 2. As a CyberPhish user, If the new email has been displayed in my inbox, then I should receive a notification with a sound indicating that a new phishing email has arrived.
GP8-24	As a team member, I want to integrate the chosen model with the app, so that the received emails are analyzed.	3	knowledge acquisition	Done	1. The model has been integrated with the app. 2. The integrated model analyzes received emails.
GP8-25	As a CyberPhish user, I want the links on my received emails to be analyzed, so that I can detect phishing URLs.	3	Feature	Done	1. As a CyberPhish user, If I receive an email with links, then all the links in the email should be analyzed.

GP8-27	As a CyberPhish user, I want the sender's email to be checked so that it can be distinguished as a legitimate or phishy sender.	2	Feature	Done	1. As a CyberPhish user, If I receive an email, then the sender's email should be checked.
GP8-28	As a CyberPhish user, I want to choose the preferred time frame for the report (year, month, week), So that I can track the report easily.	5	Feature	Done	1. As a CyberPhish user, If I enter the report page, then I should be able to choose the desired timeframe.
GP8-29	As a CyberPhish user, I want to view a statistical analysis about the received emails in the current week, so that I can track my email inbox status during this week.	3	Feature	Done	1. As a CyberPhish user, If I enter the report page, then I should be able to choose this week button. 2. As a CyberPhish user, If I click on this week button, then I should be able to view analysis of my previous received emails during the current week.
GP8-30	As a CyberPhish user, I want to view a statistical analysis about the received emails in the current month, so that I can track my email inbox status during this month.	3	Feature	Done	1. As a CyberPhish user, If I enter the report page, then I should be able to choose this month button. 2. As a CyberPhish user, If I click on this month button, then I should be able to view analysis of my previous received emails during the current month.
GP8-31	As a CyberPhish user, I want to view a statistical analysis about the received emails in the current year, so that I can track my email inbox status during this year.	3	Feature	Done	1. As a CyberPhish user, If I enter the report page, then I should be able to choose this year button. 2. As a CyberPhish user, If I click on this week button, then I should be able to view my previous received emails during the current year.

GP8-32	As a CyberPhish user, I want to view a linear chart about the number of phishing and legitimately received emails during specified timeframe, so that I can track my email inbox status during the specified timeframe.	3	Feature	Done	1. As a CyberPhish user, If I enter the report page, then I should be able to view the linear chart about the number of phishing and legitimately received emails in a specified timeframe.
GP8-33	As a CyberPhish user, I want to view insights about the number of the received phishing emails during a specified timeframe, so that I can be aware of the total number of the received phishing emails during this timeframe.	3	Feature	Done	1. As a CyberPhish user, If I enter the report page, then I should be able to view the insights about the number of phishing and legitimate received emails in a specified timeframe.
GP8-34	As a CyberPhish user, I want to view insights about the most phishy sender during a specified timeframe, so that I can be aware of this risky sender.		Feature	Done	1. As a CyberPhish user, If I enter the report page, then I should be able to view the insights about the riskiest sender during the specified timeframe.
GP8-35	As a CyberPhish user, I want to view a pie chart about the most triggering aspect of the three aspects: links, language, and sender reputation, so that I can know the most triggering aspect.	3	Feature	Done	1. As a CyberPhish user, If I enter the report page, then I should be able to view the pie chart about the most triggering aspect during the specified timeframe.
GP8-36	As a CyberPhish user, I want to filter my inbox, so that I can down narrow my inbox list of choices.	3	Feature	Done	1. As a CyberPhish user, If I was properly signed in, then I should be able to view the filter icon in my inbox. 2. As a CyberPhish user, If I choose a category from the filter, then my inbox should have only the chosen category.
GP8-37	As a CyberPhish user, I want the application to be available 99% of the time, so that I will not miss a chance to detect a phishing email.		Non-functional	Done	1. The application should be available when the user launches it.

GP8-38	As a CyberPhish user, I want the feedback to be displayed with no delay having a stable internet connection, so that I don't waste my time.		Non-functional	Done	1. The feedback of an email should be displayed during ten seconds after analyzing it.
GP8-39	As a CyberPhish user, I want the application to be simple to use, so that I do not make mistakes and waste time and energy learning how to use the application.		Non-functional	Done	1. The user should be able to use the application with no errors.
GP8-40	As a CyberPhish user, I want the application's performance to be consistent, so that no matter how many emails have been examined while using it, the system won't crash.		Non-functional	Done	1. The application's performance should be consistent within the average time range. 2. The average time for analyzing an email should not exceed 10 seconds.
GP8-41	As a CyberPhish user, I want my sessions to be inactivated and destroyed after I log out, so that no one can hijack my session.		Non-functional	Done	1. The user's session and account should be deleted once the user logs out.

4.3 System Design

In this chapter the system analysis and design will be covered. It is a creative activity in which we identify the software components of CyberPhish, as well as the relationships based on the system requirements of the CyberPhish user.

4.3.1 Architectural Diagram

The application uses a similar concept to a client-server architecture as shown in Figure 10. The architecture of a computer in which many clients request and receive service from a centralized server (host computer). Client computers provide an interface to allow a computer user to request services from the server and to display the results the server returns [29]. Within this type of model, more clients and servers can be embedded into the server, which makes the performance outstanding and increases the model's overall flexibility, in addition to the model's efficiency in delivering resources to the client while also requiring low-cost maintenance [30].

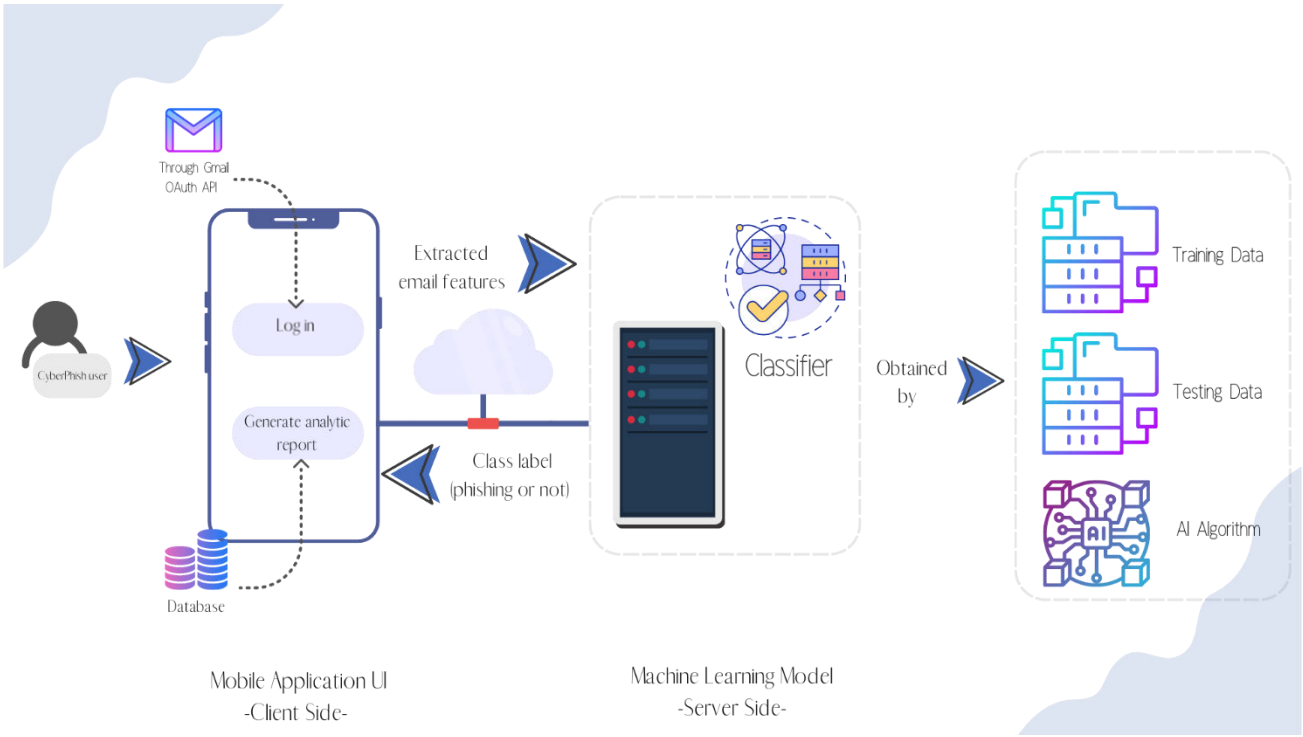


Figure 10: Architecture Diagram in 4.3.1

As shown in the above architecture, users of CyberPhish can sign in using their Gmail account through Gmail's API and interact with the application's user interface. This will enable the app to see the user's inbox on a read-only basis before extracting it. The network will be used to send the extracted email data to the server side, where the machine learning model exists.

The machine learning model is built using algorithms the SVM algorithm. The model aims to analyze the email and classify whether it is phishing or not. The server will deliver the class label and the indicators of phishing, if any, as an outcome of the classification process. These results will be made available to the user via the user interface. If logged in, the user can also request to see the analytics for their report history. This necessitates that the program maintains a record of the users' database-stored labeled emails.

4.3.2 Class Diagram /DFD

In this section the flow of data through the processes and the system and information about the outputs and inputs of each entity and the process itself as illustrated in Figure 11 will be shown.

In the dataflow diagram of CyberPhish, we start the flow of data from the login page. The data including authentication credentials go to the Gmail API. From the Gmail API the

account information data goes to the Firebase to be stored. The Gmail API then returns the account information to the CyberPhish user home. The display screen data goes from the login page to the CyberPhish user home when user gets logged in. Another function, the extracted emails data goes from the Gmail API to the CyberPhish user home. From the CyberPhish user home the display screen data goes to the awareness content, where the content data is retrieved from Firebase. The CyberPhish user home sends the display screen data to the report page.

The display screen data can go from the awareness content, the report page, or the CyberPhish user home to the chatbot. Dialogflow and Kommunicate send intent type and response type respectively to the chatbot. The report data is sent from Firebase, which takes the insight data from API Void, and is sent to the report page. At the end, the log out data, can be sent from the CyberPhish user home, the report page, or the awareness content page which returns the user to login page.

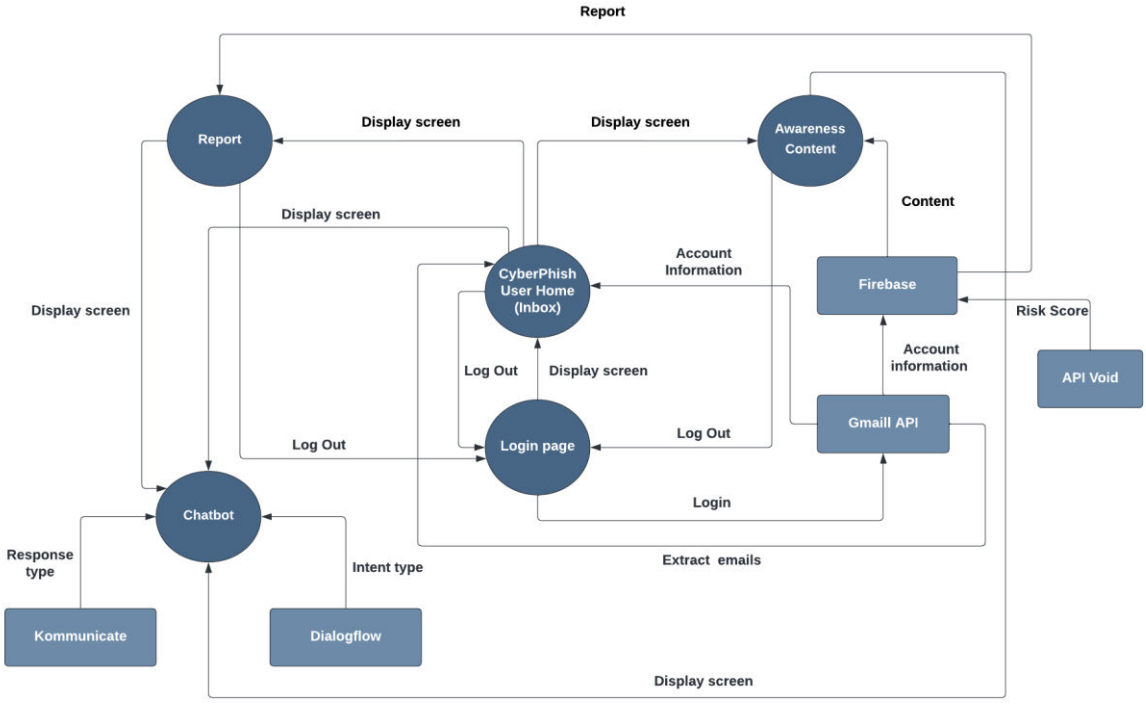


Figure 11: Data flow diagram in 5.2

4.3.3 Component Level Design

This section showcases how each component in the CyberPhish application is implemented, using UML diagrams, activity diagrams.

- UML diagram:

The UML diagram of CyberPhish represents the different relationships between classes, the attributes and methods each class has, and the multiplicities between them. As shown in Figure 12, the WelcomeScreen and the LoginScreen are both interface classes, having a one-to-one relationship on both sides. Then the LoginScreen has a one-to-one relationship with both the LoginBackend and MyElevatedButton classes. After that, the LoginScreen calls the LoginBackend to start the application backend process. Also, the LoginBackend has a zero-to-many relationship with the Email object class and a one-to-many relationship with the Article object class. Furthermore, the LoginBackend calls the NavBar class and has a one-to-many relationship on both sides.

The NavBar class connects the HomeScreen, ReportScreen, and AwarenessContent screens, and it has a one-to-one relationship with all of them on both sides. As for the HomeScreen, it calls the APIBackend to start the extraction process of the emails, and they have a one-to-one relationship on both sides. The APIBackend calls the ExtractEmail class to complete the extraction process of the emails and have a one-to-one relationship on both sides. Also, the ExtractEmail class calls the NotificationBackend class to send the notification of the emails and have a one-to-one relationship on both sides.

When it comes to the end of the email extraction process, the HomeScreen displays the inbox by having a composition relationship with the MailCard. Also, the MailCard has a one-to-one relationship with the EmailScreen. The EmailScreen has a one-to-one relationship with the bodyBuilder class. Furthermore, Both the MailCard and EmailScreen have a composition relationship with the Email object class.

As for the AwarenessContent class, it has a composition relationship with the ArticleCard class. Also, the ArticleCard has a composition relationship with the Article class. Further, the Article class has a one-to-one relationship with the ArticleScreen interface.

Lastly, all of the YearDashboard, MonthDashboard, and WeekDashboard screens have an inheritance relationship with the ReportScreen, which displays the analytical report to CyberPhish's user. Also, the SizeConfig class is an object of the design of CyberPhish's application and has an association relationship with most of the classes to display a consistent design.

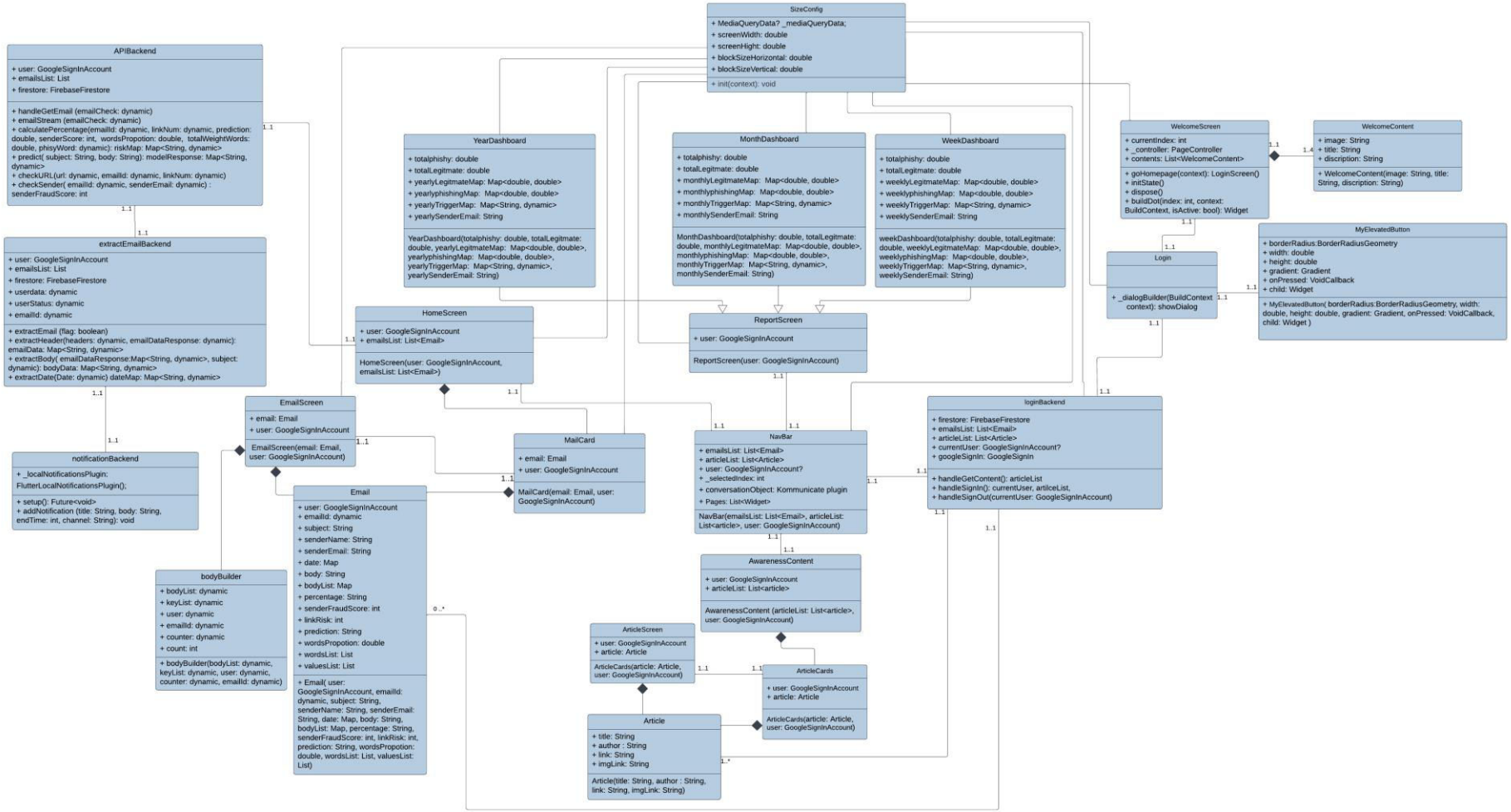


Figure 12: UML Diagram in 5.3.1

- Activity Diagram of Login:

The login activity diagram as depicted Figure 13, shows that when the user logs in through Gmail API, using their Gmail account successfully, they will be able to see the CyberPhish user home (inbox). Otherwise, the Gmail API page will ask the user to input correct credentials. If the user fails to login the cycle stops and the user begins again.

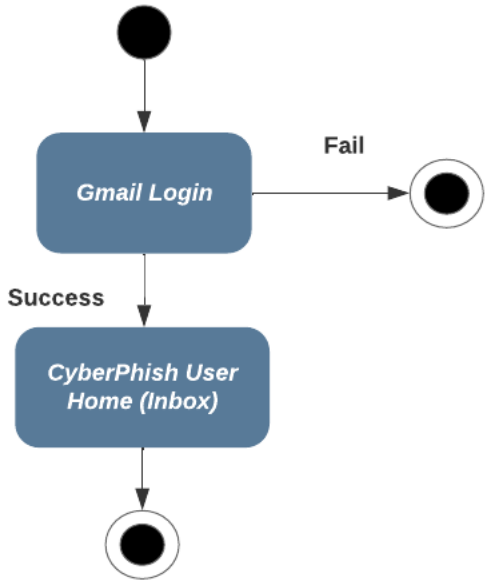


Figure 13: Login Activity diagram in 5.3.2

- Activity Diagram of View Inbox:

The view inbox activity diagram as illustrated in Figure 14, shows that when the user logs in through Gmail API, using their Gmail account successfully, they will be able to see the CyberPhish user home(inbox). Otherwise, the Gmail API page will ask the user to input correct credentials. Then, the user will be able to select a phishy or legitimate specific email to view. When the user selects a phishy email to view they will be able to view the percentage of phishy words, or the risk percentage of links in the email, or the risk percentage of the sender of that specific email. If the user fails to login the cycle stops and the user begins again.

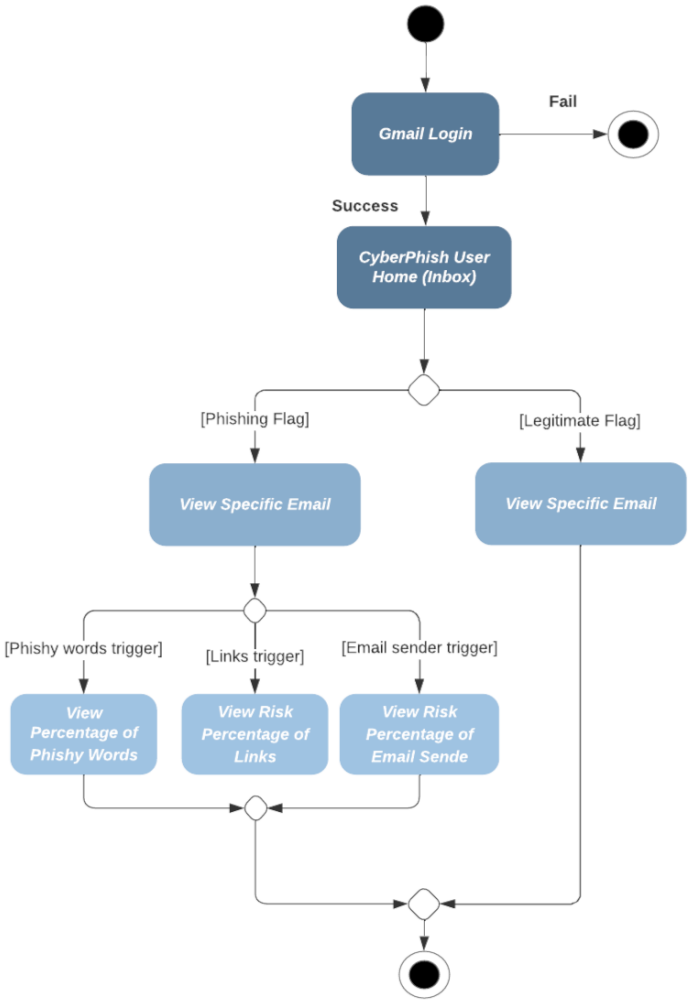


Figure 14: View Inbox activity diagram in 5.3.3

- Activity Diagram of Chatbot:

The activity diagram of the chatbot as shown in Figure 15, depicts that when the user successfully logs in through Gmail, and navigates to the home page, the user can press on the

chatbot button, and have two choices, either play a quiz game, or view options that are considered as frequently asked questions. If the user fails to login the cycle stops and the user begins again.

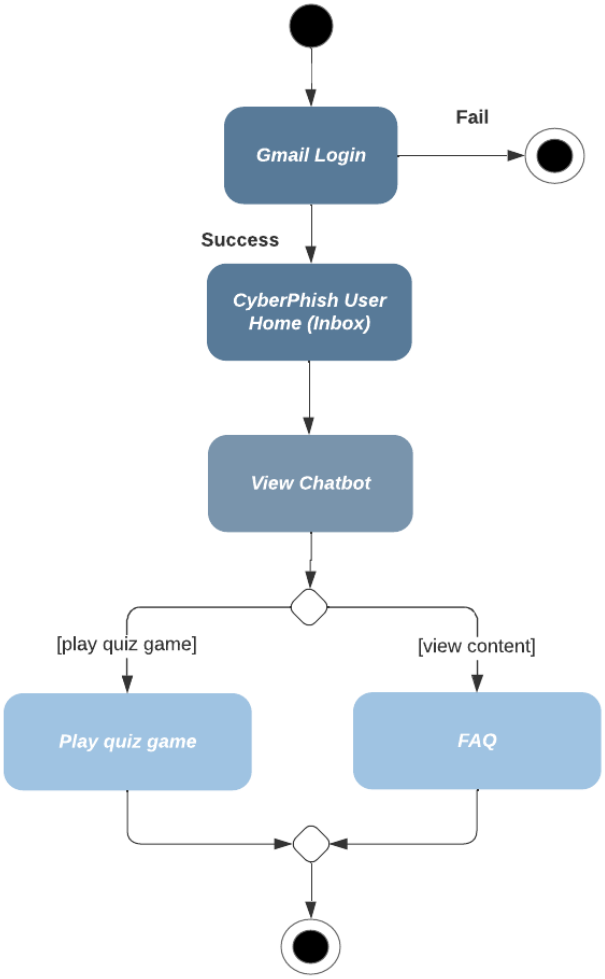


Figure 15: Chat bot Activity diagram in 5.3.4

- Activity Diagram of Awareness Content:

The awareness content activity diagram as depicted in Figure 16, illustrates that when the user logs in through Gmail API, using their Gmail account successfully, they will be able to see the CyberPhish user home(inbox). Otherwise, the Gmail API page will ask the user to

input correct credentials. Then, the user will be able to see the navigation bar, where they can tap on awareness content icon, to view the awareness content page and select a specific article to read. If the user fails to login the cycle stops and the user begins again.

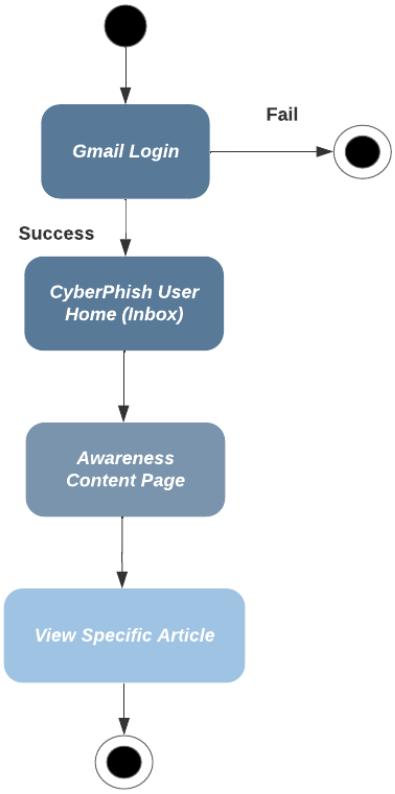


Figure 16: Awareness content activity diagram in 5.3.5

- Activity Diagram of Report Page:

The report page activity diagram as depicted in Figure 17, illustrates that when the user logs in through Gmail API, using their Gmail account successfully, they will be able to see the CyberPhish user home(inbox). Otherwise, the Gmail API page will ask the user to input correct credentials. Then, the user will be able to see the navigation bar, where they can tap on report page icon, to view the analytics report page. The user can view weekly, or

monthly, or yearly analytics report of their inbox. If the user fails to login the cycle stops and the user begins again.

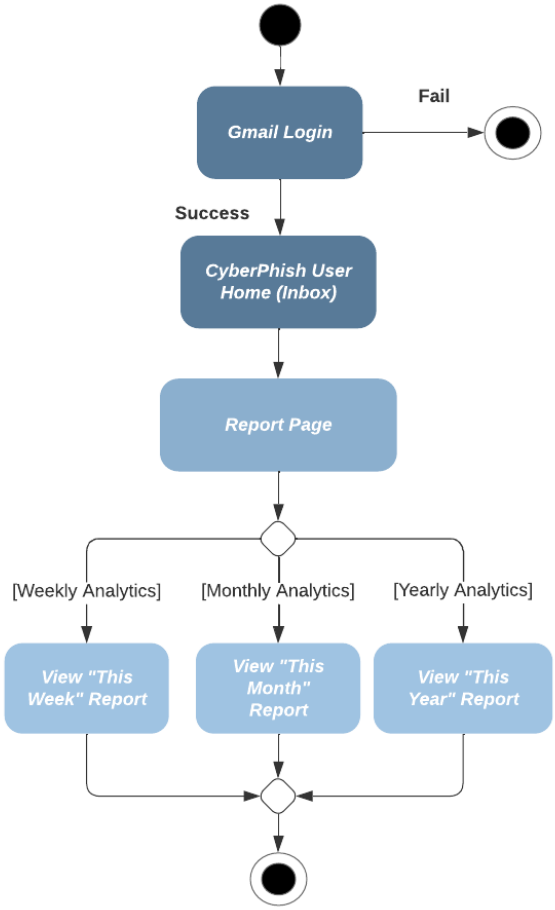


Figure 17: Report page activity diagram in 5

4.4 Data Design

This section describes and illustrates data flow in our database using ER diagrams and non-relational data model.

4.4.1 Data Models

- ER Diagram:

The ER diagram in Figure 18 shows, that each GoogleSignInAccount, has userId, email, photoURL, displayName, and userStatus. Also, the GoogleSignInAccount has only one emailsList, and the emails list contains many emails, each email has an emailId, subject, senderEmail, senderName, body, bodyList, subject, date, linkRisk, modelPrediction, numPhishyWord, totalWeightWords, percentage, prediction, senderRiskScore, trigger, wordsList, valuesList, and wordProportion. Moreover, the email entity has a links list, with each link having LinkString and RiskScore. The GoogleSignInAccount has one report with a one-year list. The year entity has two attributes the totalYear and the month and it has one or more months. The month entity has the totalM and the weeks attributes and has one or more weeks. The week entity has the totalW and days attributes. Lastly, the AwarenessContent is a stand-alone entity that has four attributes: title, author, link, and imgLink.

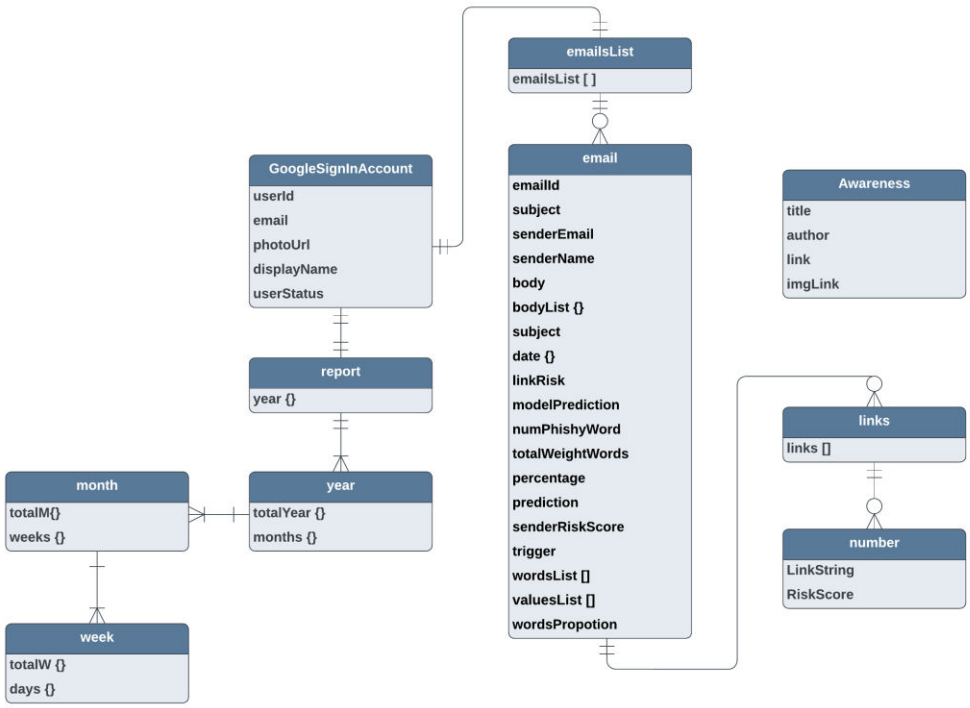


Figure 18: ER Diagram in 5.4.1.1

- Non-relational data model:

The Non-relation data model as illustrated in Figure 19 shows that CyberPhish has a Firestore cloud database, where it has a GoogleSignInAccount collection, that has a document defined by the userId. This document has the userId attribute. The GoogleSignInAccount collection has two other collections inside it, the first is called EmailsList. The EmailsList collection has many documents, which contain the email data. The email document contains the following attributes: emailld, subject, senderEmail, senderName, body, bodyList, subject, date, linkRisk, modelPrediction, numPhishyWord, totalWeightWords, percentage, prediction, senderRiskScore, trigger, wordsList, valuesList, wordsPropotion, and wordsList. The second collection inside the GoogleSignInAccount is the Report. The report is made up of nested collections starting from the current year which has the totalYear2023 map that has the legitimate, phishing, and senderEmail attributes. Then, a nested map inside the totalYear2023 which is call the triggersMap which has the language, sender, and the link attribute. These two maps are repeated for each timeframe. Lastly the AwarenessContent collection which has the articles as documents. Each article document has the title, author, link, and imgLink attributes.

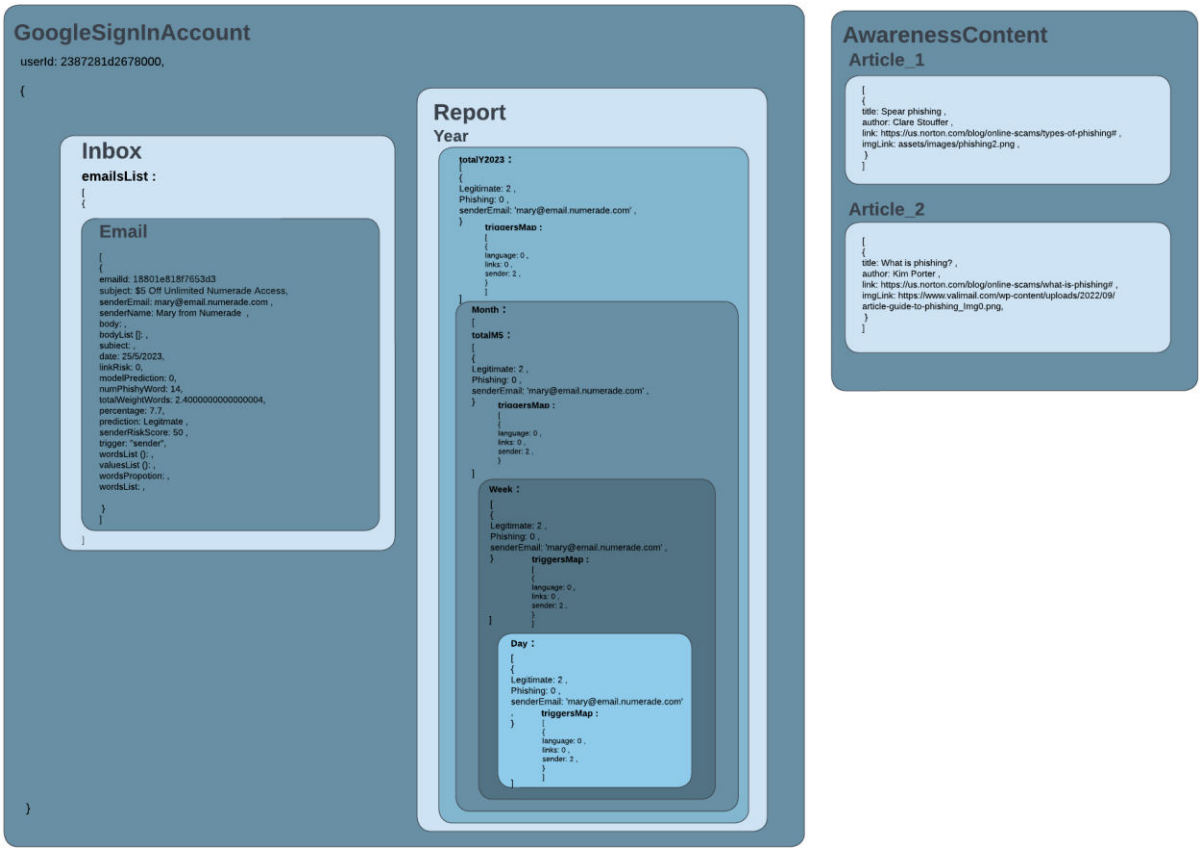
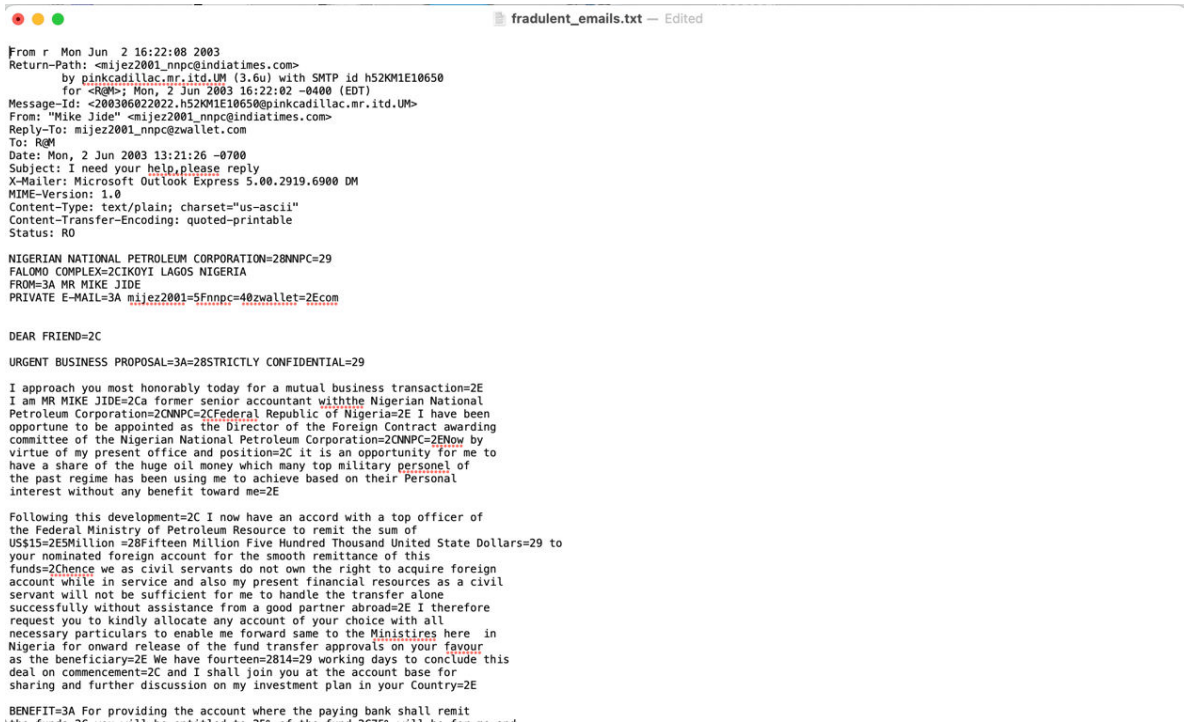


Figure 19: Non- relational Data Model in 5.4.1.2

4.4.2 Data Collection and Preparation

For data involved with CyberPhish the team members have searched for datasets through many resources like UCI, Google Scholar, GitHub, Kaggle, and Harvard data verse. The team looked for datasets that were complete, and with the required attributes, to ensure that the data collected is of highest quality. As a result, many datasets were found, like the dataset Spam and ham which included emails. Those emails were lacking the sufficient attributes only having subject, body, and labeled either spam or ham (i.e., not spam). Moreover, the dataset of phishing emails that was thought to be suitable. This dataset had the attributes including the sender, body, message ID, however it had only 190 tuples which has not been enough. One of the datasets stumbled upon was of both phishing and legitimate emails, but the dataset was fully encoded with no way to know or ask about the meaning of each attribute. Another dataset that was promising, with many tuples, but the dataset was private, meaning it could not be downloaded, and there was no email address of the owner to ask for permission. Also, the Enron dataset that has all the needed attributes, with about 5000 tuples, but the significant drawback was that it includes a company domain, meaning that it is going to be biased, which is not the project's target.

After the thorough search process, a dataset in Kaggle stood out. This dataset had fraudulent emails in txt format, containing 3916 tuples with no class label. To extract the desired data from the text file a Python code was implemented, where extraction of each column data each column's data and write it in a CSV format file to be used later, then added a class label (phishing) as shown in Figure 20.



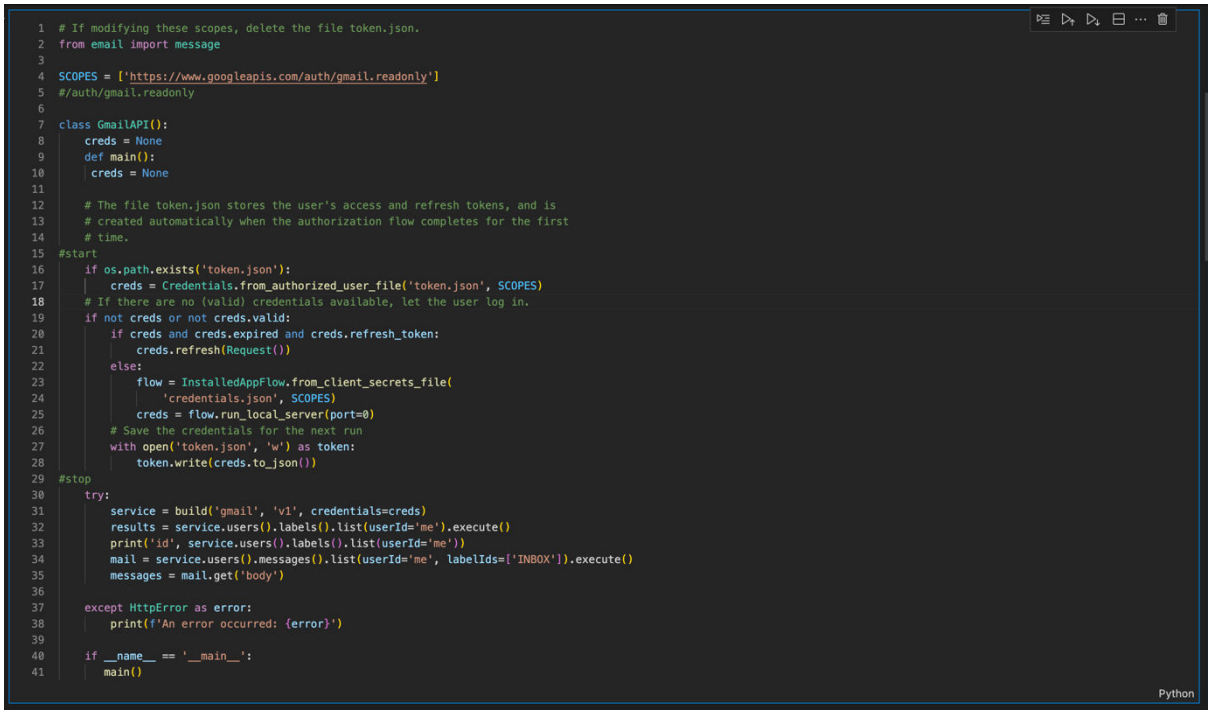
fradulent_emails.txt — Edited

```
From r  Mon Jun  2 16:22:08 2003
Return-Path: <mijez2001_nnpc@indiatimes.com>
        by pinkcadillac.mr.itd.UM (3.6u) with SMTP id h52KM1E10650
        for <R@M>; Mon, 2 Jun 2003 16:22:02 -0400 (EDT)
Message-Id: <200306022022.h52KM1E10650@pinkcadillac.mr.itd.UM>
From: "Mike Jide" <mijez2001_nnpc@indiatimes.com>
Reply-To: mijez2001_nnpc@zwallet.com
To: R@M
Date: Mon, 2 Jun 2003 13:21:26 -0700
Subject: I need your help,please reply
X-Mailer: Microsoft Outlook Express 5.00.2919.6900 DM
MIME-Version: 1.0
Content-Type: text/plain; charset="us-ascii"
Content-Transfer-Encoding: quoted-printable
Status: RO

NIGERIAN NATIONAL PETROLEUM CORPORATION=28NNPC=29
FALOMO COMPLEX=2CIKOYI LAGOS NIGERIA
FROM=3A MR MIKE JIDE
PRIVATE E-MAIL=3A mijez2001=5Fnnpc=40zwallet=2Ecom


DEAR FRIEND=2C

URGENT BUSINESS PROPOSAL=3A=28STRICTLY CONFIDENTIAL=29

I approach you most honorably today for a mutual business transaction=2E
I am MR MIKE JIDE=2Ca former senior accountant withthe Nigerian National
Petroleum Corporation=2CNNPC=2CFederal Republic of Nigeria=2E I have been
opportune to be appointed as the Director of the Foreign Contract awarding
committee of the Nigerian National Petroleum Corporation=2CNNPC=2ENow by
virtue of my present office and position=2C it is an opportunity for me to
have a share of the huge oil money which many top military personel of
the past regime has been using me to achieve based on their Personal
interest without any benefit toward me=2E

Following this development=2C I now have an accord with a top officer of
the Federal Ministry of Petroleum Resource to remit the sum of
US$15=2E5Million =28Fifteen Million Five Hundred Thousand United State Dollars=29 to
your nominated foreign account for the smooth remittance of this
funds=2Chence we as civil servants do not own the right to acquire foreign
account while in service and also my present financial resources as a civil
servant will not be sufficient for me to handle the transfer alone
successfully without assistance from a good partner abroad=2E I therefore
request you to kindly allocate any account of your choice with all
necessary particulars to enable me forward same to the Ministires here  in
Nigeria for onward release of the fund transfer approvals on your favour
as the beneficiary=2E We have fourteen=2814=29 working days to conclude this
deal on commencement=2C and I shall join you at the account base for
sharing and further discussion on my investment plan in your Country=2E

BENEFIT=3A For providing the account where the paying bank shall remit
```

	Message-ID	From	Subject	BODY	Label
2	<200306022022.	"Mike Jide" <I	need your h	by pinkcadill	phishing
3	<Sea2-F88CfCsvC	"abram nkon	YOUR URGEN	by magnumf	phishing
4	<3026278.10552	maira ssesko	CRY FOR HEL	<0HG900LR(	phishing
5	<1055288816.158487.5917.zr	ASSITANCE	ABUGAL & C(	phishing
6	<1055275912.991633.46365.:	GOD'S WORI	Beloved in Ch	phishing
7	<2003061106564	Joseph Mobu	CONFIDENTI	This is a mult	phishing
8	<200306111552.	"Mariam Aba	Can You Keep	A very Good	phishing
9	<1055469288.927394.79429.:	Urgent respo	Union Bank c	phishing
10	<2003061311111	Joseph Mobu	CONFIDENTI	This is a mult	phishing
11	<200306130724.	"PRINCE FOL	TREAT AS UR	DEAR SIR=2	phishing
12	<1055520682015	sussy emma	GOODDAY SI	This message	phishing
13	<200306141229.	"Harry O Han	seeking for y	CHIEF HARRY	phishing
14	<200306141139.	Drmoore Jam	Please reply	DR. JAMES M	phishing
15	<200306141731.	"PRINCE WIL	THIS IS PRIV	--====_SecAtt	phishing
16	<200306141751.	"PRINCE M	\THIS IS PRIV	--====_SecAtt	phishing
17	<200306142330.	"MRS CECILI	PLEASE GET E	Dear Good Fr	phishing
18	<200306161152.	MECK MAHA	WINNING N	This is a mult	phishing
19	<20030616122048.19914.qm:	Business Par	DEAR CEO/D	phishing
20	<200306161434.	"FRANK BELL	HELP/BUSINI	From=3AFRA	phishing
21	<200306161434.	"FRANK BELL	HELP/BUSINI	From=3AFRA	phishing
22	<200306162321.	"Mr. Mike Ob	MY FRIEND, I	This is a mult	phishing
23	<200306170120.	"Barrister. Ba	URGENT ASS	DEAR SIR=2C	phishing
24	<E19TDFO-0003T	"MUYIWA IGE" <bechman	Dear Sir=2FM	phishing
25	<127.0.0.1+Tt0pXbxD4nCdnW	MRS SARA LA	Dear , I know	phishing
26	<200306232313.	"Dr.Tunde Co	With Utmost	Investment p	phishing
27	<200306240026.	"Dr Tunde Co	With Utmost	Investment p	phishing
28	<200306241210.	"williamume'	fund transfe	Dear Sir=2CM	phishing
29	<200306242212	"Mr Philip Io	please reply	(Executive M	phishing

Figure 20: Conversion from TXT to CSV in 5.4.2

Since no dataset with legitimate emails was available, using the Python code and the Gmail API, a dataset from extracted emails that was certainly not phishing from our own Gmail accounts was created to balance the fraudulent dataset. Gmail API was used to gain access to the emails in the inbox and extract them as viewed in Figure 21.



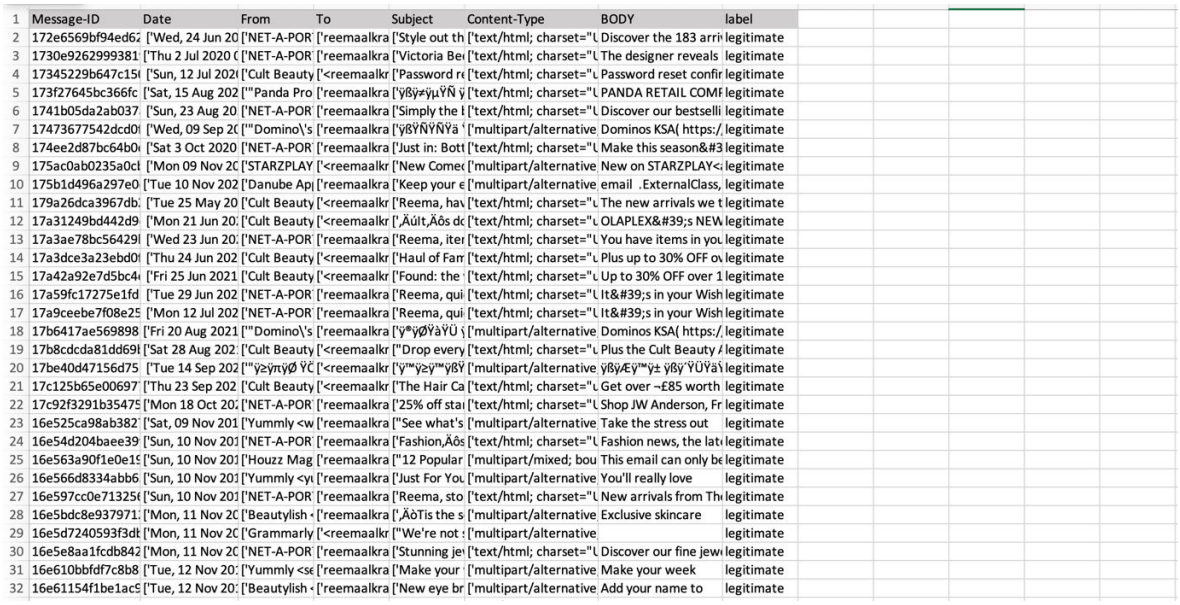
```python
# If modifying these scopes, delete the file token.json.
from email import message

SCOPES = ['https://www.googleapis.com/auth/gmail.readonly']
#/auth/gmail.readonly

class GmailAPI():
    creds = None
    def main():
     creds = None

    # The file token.json stores the user's access and refresh tokens, and is
    # created automatically when the authorization flow completes for the first
    # time.
#start
    if os.path.exists('token.json'):
        creds = Credentials.from_authorized_user_file('token.json', SCOPES)
    # If there are no (valid) credentials available, let the user log in.
    if not creds or not creds.valid:
        if creds and creds.expired and creds.refresh_token:
            creds.refresh(Request())
        else:
            flow = InstalledAppFlow.from_client_secrets_file(
                'credentials.json', SCOPES)
            creds = flow.run_local_server(port=0)
        # Save the credentials for the next run
        with open('token.json', 'w') as token:
            token.write(creds.to_json())
#stop
    try:
        service = build('gmail', 'v1', credentials=creds)
        results = service.users().labels().list(userId='me').execute()
        print('id', service.users().labels().list(userId='me'))
        mail = service.users().messages().list(userId='me', labelIds=['INBOX']).execute()
        messages = mail.get('body')

    except HttpError as error:
        print(f'An error occurred: {error}')

    if __name__ == '__main__':
        main()
```

Figure 21: Python Gmail API in 5.4.2

Due to using the Gmail API as students and not an enterprise, we were only able to extract 100 emails at a time; therefore, the extraction process had to work in a loop where we extract from the inbox, move extracted emails out of the inbox manually, extract the new emails from the inbox, then append all the extracted emails in another CSV format file, resulting in 2014 tuples, and add a class label to them (legitimate) as shown in Figure 22. The emails we extracted from our Gmail accounts were not enough, so we had to ask volunteers for their Gmail accounts and repeat the extraction process on their emails, resulting in a total of 3121 tuples as seen in Figure 23

```python
gmail_parser= GmailParser()
gmail_service= GmailAPI()

mail = gmail_service.service.users().messages().list(userId='me', labelIds=['INBOX']).execute()
messages = mail.get('messages')
header = ['Message-ID' , 'Date' , 'From' , 'To', 'Subject' , 'Content-Type','BODY']

with open('data_collectNP_CP.csv', 'a', encoding='UTF8') as w:
    writer = csv.writer(w)
    writer.writerow(header)
    for e in messages:
     message = gmail_service.service.users().messages().get(userId='me', id=e['id'], format="full").execute()
     message_id=message['id']
     messageheader= gmail_service.service.users().messages().get(userId="me", id=e['id'], format="full", metadataHeaders=None).execute()
     headers=messageheader["payload"]["headers"]
     subject= [i['value'] for i in headers if i["name"]=="Subject"]
     From= [i['value'] for i in headers if i["name"]=="From"]
     Date= [i['value'] for i in headers if i["name"]=="Date"]
     Content_Type= [i['value'] for i in headers if i["name"]=="Content-Type"]
     to=[i['value'] for i in headers if i["name"]=="To"]

     body = gmail_parser.read_message(content=message)
     record=[message_id,Date, From ,to, subject,Content_Type,body]

     writer.writerow(record)
```

Figure 22: Extracting email data & writing in CSV file in 5.4.2



	Message-ID	Date	From	To	Subject	Content-Type	BODY	label
2	172e6569bf94ed67	['Wed, 24 Jun 20	['NET-A-POR	['reemaalkra	['Style out th	['text/html; charset="U	Discover the 183 arri	legitimate
3	1730e9262999381	['Thu 2 Jul 2020 (	['NET-A-POR	['reemaalkra	['Victoria Be	['text/html; charset="U	The designer reveals	legitimate
4	17345229b647c15	['Sun, 12 Jul 202	['Cult Beauty	['<reemaalkr	['Password re	['text/html; charset="U	Password reset confir	legitimate
5	173f27645bc366fc	['Sat, 15 Aug 202	['"Panda Pro	['reemaalkra	['ÿßÿ≠ÿµŸÑ ÿ	['text/html; charset="U	PANDA RETAIL COMF	legitimate
6	1741b05da2ab037	['Sun, 23 Aug 20	['NET-A-POR	['reemaalkra	['Simply the I	['text/html; charset="U	Discover our bestselli	legitimate
7	17473677542dcd0	['Wed, 09 Sep 2(	['"Domino\'s	['reemaalkra	['ÿßŸÑŸÑŸà '	['multipart/alternative	Dominos KSA(https:/	legitimate
8	174ee2d87bc64b0	['Sat 3 Oct 2020	['NET-A-POR	['reemaalkra	['Just in: Bott	['text/html; charset="U	Make this season&#3	legitimate
9	175ac0ab0235a0cl	['Mon 09 Nov 2(	['STARZPLAY	['<reemaalkr	['New Comec	['multipart/alternative	New on STARZPLAY<;	legitimate
10	175b1d496a297e0	['Tue 10 Nov 202	['Danube Ap	['reemaalkra	['Keep your e	['multipart/alternative	email .ExternalClass,	legitimate
11	179a26dca3967db	['Tue 25 May 20	['Cult Beauty	['<reemaalkr	['Reema, hav	['text/html; charset="U	The new arrivals we t	legitimate
12	17a31249bd442d9	['Mon 21 Jun 20:	['Cult Beauty	['<reemaalkr	['‚Äúlt‚Äôs dc	['text/html; charset="U	OLAPLEX's NEW	legitimate
13	17a3ae78bc56429	['Wed 23 Jun 20:	['NET-A-POR	['reemaalkra	['Reema, iter	['text/html; charset="U	You have items in you	legitimate
14	17a3dce3a23ebd0	['Thu 24 Jun 202	['Cult Beauty	['<reemaalkr	['Haul of Fam	['text/html; charset="U	Plus up to 30% OFF ov	legitimate
15	17a42a92e7d5bc4	['Fri 25 Jun 2021	['Cult Beauty	['<reemaalkr	['Found: the '	['text/html; charset="U	Up to 30% OFF over 1	legitimate
16	17a59fc17275e1fd	['Tue 29 Jun 202	['NET-A-POR	['reemaalkra	['Reema, qui	['text/html; charset="U	It's in your Wish	legitimate
17	17a9ceebe7f08e25	['Mon 12 Jul 202	['NET-A-POR	['reemaalkra	['Reema, qui	['text/html; charset="U	It's in your Wish	legitimate
18	17b6417ae569898	['Fri 20 Aug 2021	['"Domino\'s	['reemaalkra	['ÿ*ÿØŸàŸÜ ÿ	['multipart/alternative	Dominos KSA(https:/	legitimate
19	17b8cdcda81dd69	['Sat 28 Aug 202	['Cult Beauty	['<reemaalkr	["Drop every	['text/html; charset="U	Plus the Cult Beauty A	legitimate
20	17be40d47156d75	['Tue 14 Sep 202	['"ŸƵ≥ÿπÿØ ŸČ	['<reemaalkr	['ŸˆŸ™ŸƵ≥Ÿ™ÿßŸ	['multipart/alternative	ÿßŸÆŸ™Ÿ± ÿßŸˆŸÜŸàŸ	legitimate
21	17c125b65e00697	['Thu 23 Sep 202	['Cult Beauty	['<reemaalkr	['The Hair Ca	['text/html; charset="U	Get over ¬£85 worth	legitimate
22	17c92f3291b35475	['Mon 18 Oct 20:	['NET-A-POR	['reemaalkra	['25% off stal	['text/html; charset="U	Shop JW Anderson, Fr	legitimate
23	16e525ca98ab382	['Sat, 09 Nov 201	['Yummly <w	['reemaalkra	['See what's	['multipart/alternative	Take the stress out	legitimate
24	16e54d204baee39	['Sun, 10 Nov 201	['NET-A-POR	['reemaalkra	['Fashion,Âôs	['text/html; charset="U	Fashion news, the lat	legitimate
25	16e563a90f1e0e15	['Sun, 10 Nov 201	['Houzz Mag	['reemaalkra	["12 Popular	['multipart/mixed; bou	This email can only be	legitimate
26	16e566d8334abb6	['Sun, 10 Nov 201	['Yummly <yu	['reemaalkra	['Just For You	['multipart/alternative	You'll really love	legitimate
27	16e597cc0e71325	['Sun, 10 Nov 201	['NET-A-POR	['reemaalkra	['Reema, sto	['text/html; charset="U	New arrivals from Th	legitimate
28	16e5bdc8e937971	['Mon, 11 Nov 2(	['Beautylish <	['reemaalkra	['‚ÄòTis the s	['multipart/alternative	Exclusive skincare	legitimate
29	16e5d7240593f3d	['Mon, 11 Nov 2(	['Grammarly	['<reemaalkr	["We're not s	['multipart/alternative		legitimate
30	16e5e8aa1fcdb842	['Mon, 11 Nov 2(	['NET-A-POR	['reemaalkra	['Stunning jev	['text/html; charset="U	Discover our fine jew	legitimate
31	16e610bbfdf7c8b8	['Tue, 12 Nov 20:	['Yummly <se	['reemaalkra	['Make your '	['multipart/alternative	Make your week	legitimate
32	16e61154f1be1ac9	['Tue, 12 Nov 20:	['Beautylish <	['reemaalkra	['New eye br	['multipart/alternative	Add your name to	legitimate

Figure 23: Email extraction from Gmail accounts in 5.4.2

The email sending process was tried via a Python code where you input the sender's email, their password, and the receiver's email, but it didn't work so it had to be done manually, making sure that the email bodies were from legit emails found online as viewed in Figure 24.

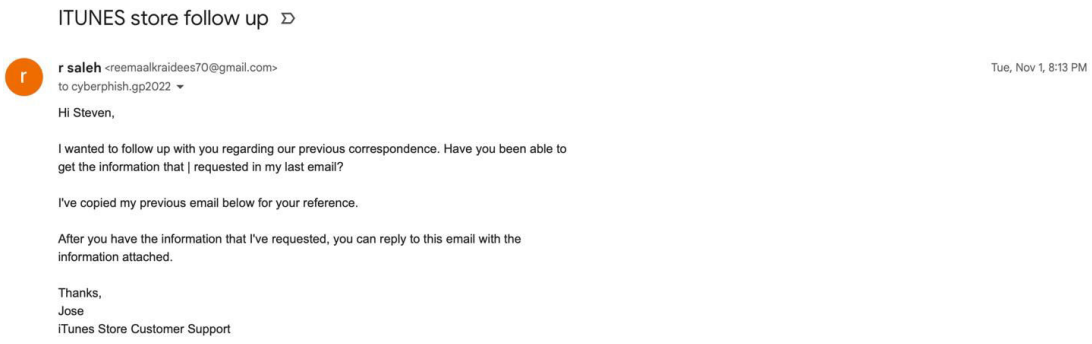
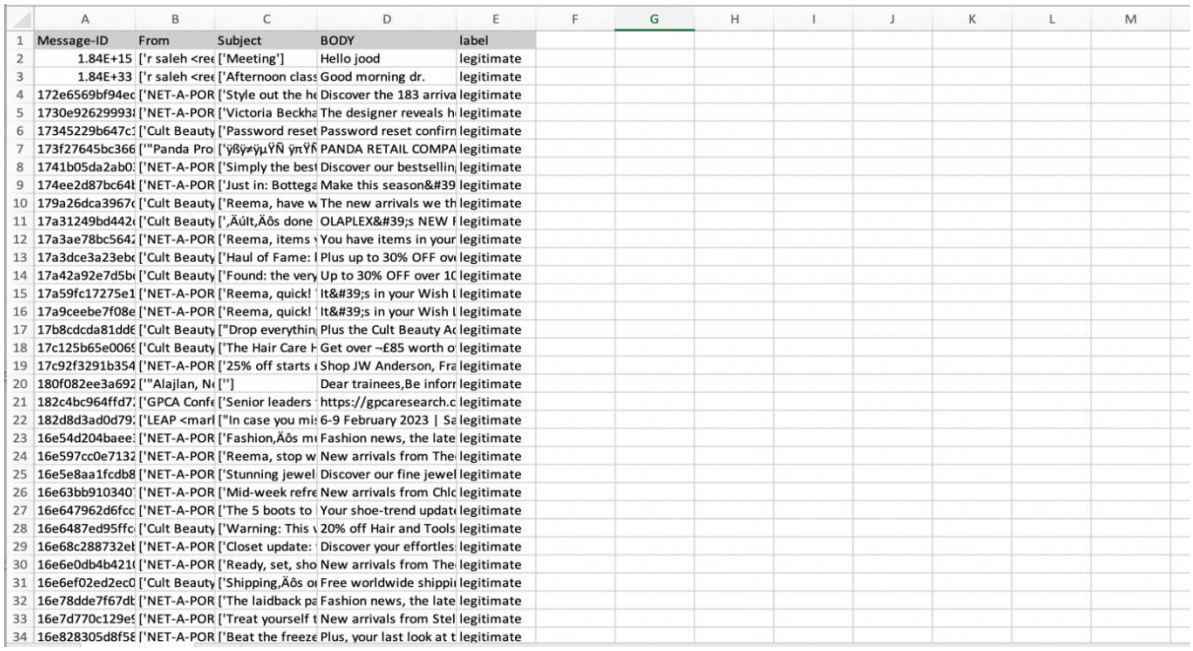


Figure 24: Manual email writing in 5.4.2

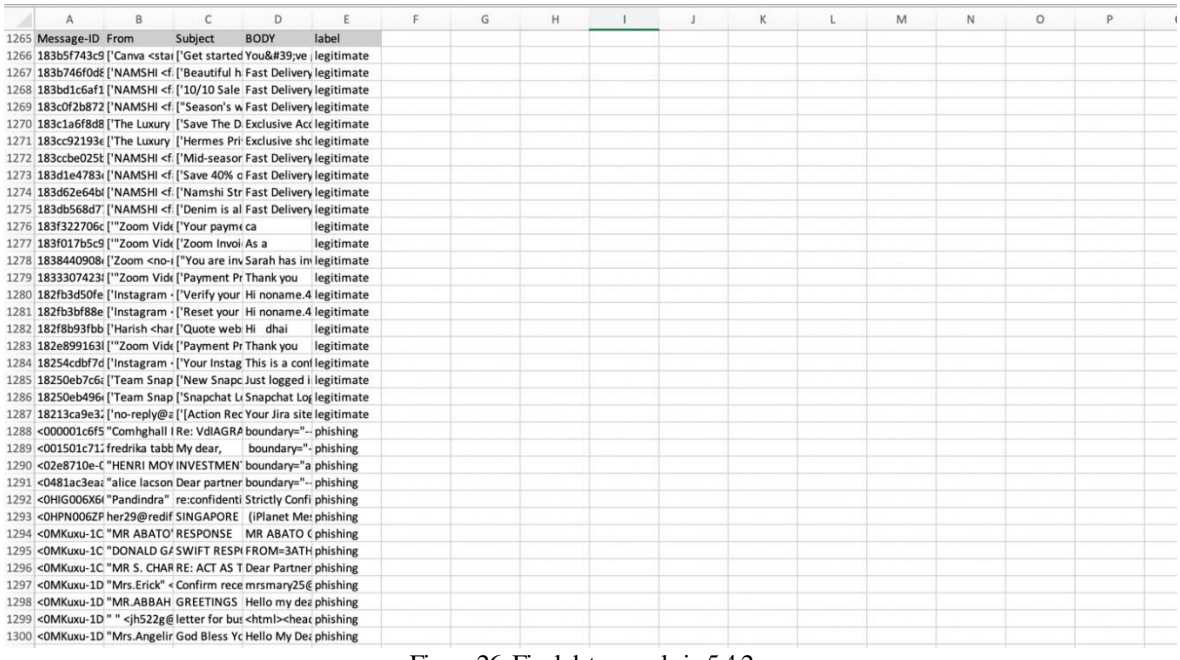
After successful data creation and collection, we started data preprocessing in Python. First, we removed duplicate and missing rows keeping only the original row. We also removed the Date and Content type column to balance the two datasets, since the phishing dataset did not have those attributes. As for outliers we didn't have any since each row is unique meaning no abnormalities from other values. As for encoding, our dataset did not need any encoding since it was all unique emails with different texts, which is our AI technique's focus. Therefore, it did not make sense to transform the emails into encoded form as illustrated in Figure 25.



	A	B	C	D	E
1	Message-ID	From	Subject	BODY	label
2	1.84E+15	['r saleh <rea	['Meeting']	Hello jood	legitimate
3	1.84E+33	['r saleh <rea	['Afternoon class	Good morning dr.	legitimate
4	172e6569bf94ec	['NET-A-POR	['Style out the he	Discover the 183 arriva	legitimate
5	1730e926299938	['NET-A-POR	['Victoria Beckha	The designer reveals h	legitimate
6	17345229b647c1	['Cult Beauty	['Password reset	Password reset confirm	legitimate
7	173f27645bc366	['"Panda Pro	['ÿßÿ≠ÿµÙÑ ÿπÙÑ	PANDA RETAIL COMPA	legitimate
8	1741b05da2ab0	['NET-A-POR	['Simply the best	Discover our bestsellin	legitimate
9	174ee2d87bc64b	['NET-A-POR	['Just in: Bottega	Make this season&#39	legitimate
10	179a26dca3967c	['Cult Beauty	['Reema, have w	The new arrivals we th	legitimate
11	17a31249bd442c	['Cult Beauty	['Ãúlt;Ãôs done	OLAPLEX's NEW F	legitimate
12	17a3ae78bc5642	['NET-A-POR	['Reema, items v	You have items in your	legitimate
13	17a3dce3a23ebc	['Cult Beauty	['Haul of Fame: I	Plus up to 30% OFF ov	legitimate
14	17a42a92e7d5b0	['Cult Beauty	['Found: the very	Up to 30% OFF over 10	legitimate
15	17a59fc17275e1	['NET-A-POR	['Reema, quick!	It's in your Wish l	legitimate
16	17a9ceebe7f08e	['NET-A-POR	['Reema, quick!	It's in your Wish l	legitimate
17	17b8cdcda81dd6	['Cult Beauty	["Drop everythin	Plus the Cult Beauty Ac	legitimate
18	17c125b65e0069	['Cult Beauty	['The Hair Care H	Get over ¬£85 worth o	legitimate
19	17c92f3291b354	['NET-A-POR	['25% off starts t	Shop JW Anderson, Fra	legitimate
20	180f082ee3a692	['"Alajlan, N	['"]	Dear trainees,Be inforr	legitimate
21	182c4bc964ffd7	['GPCA Confe	['Senior leaders	https://gpcaresearch.c	legitimate
22	182d8d3ad0d79	['LEAP <marl	['"In case you mis	6-9 February 2023 | Sa	legitimate
23	16e54d204baee	['NET-A-POR	['Fashion,Ãôs mi	Fashion news, the late	legitimate
24	16e597cc0e7132	['NET-A-POR	['Reema, stop w	New arrivals from The	legitimate
25	16e5e8aa1fcdb8	['NET-A-POR	['Stunning jewel	Discover our fine jewel	legitimate
26	16e63bb910340	['NET-A-POR	['Mid-week refre	New arrivals from Chlo	legitimate
27	16e647962d6fcc	['NET-A-POR	['The 5 boots to	Your shoe-trend updat	legitimate
28	16e6487ed95ffc	['Cult Beauty	['Warning: This v	20% off Hair and Tools	legitimate
29	16e68c288732e	['NET-A-POR	['Closet update:	Discover your effortles	legitimate
30	16e6e0db4b421	['NET-A-POR	['Ready, set, sho	New arrivals from The	legitimate
31	16e6ef02ed2ec0	['Cult Beauty	['Shipping,Ãôs o	Free worldwide shippi	legitimate
32	16e78dde7f67db	['NET-A-POR	['The laidback pa	Fashion news, the late	legitimate
33	16e7d770c129e5	['NET-A-POR	['Treat yourself t	New arrivals from Stel	legitimate
34	16e828305d8f58	['NET-A-POR	['Beat the freeze	Plus, your last look at t	legitimate

Figure 25: Legitimate data sample in 5.4.2

The resulting 1344 phishing emails and 1344 non-phishing emails were combined to result in one high quality dataset of 2688 tuples ready to be used for the training and testing processes as shown in Figure 26.



	A	B	C	D	E
1265	Message-ID	From	Subject	BODY	label
1266	183b5f743c9	['Canva <sta	['Get started	You've	legitimate
1267	183b746f0d8	['NAMSHI <f	['Beautiful h	Fast Delivery	legitimate
1268	183bd1c6af1	['NAMSHI <f	['10/10 Sale	Fast Delivery	legitimate
1269	183c0f2b872	['NAMSHI <f	["Season's w	Fast Delivery	legitimate
1270	183c1a6f8d8	['The Luxury	['Save The D	Exclusive Acc	legitimate
1271	183cc92193e	['The Luxury	['Hermes Pri	Exclusive sho	legitimate
1272	183ccbe025b	['NAMSHI <f	['Mid-seasor	Fast Delivery	legitimate
1273	183d1e4783	['NAMSHI <f	['Save 40% o	Fast Delivery	legitimate
1274	183d62e64b	['NAMSHI <f	['Namshi Str	Fast Delivery	legitimate
1275	183db568d7	['NAMSHI <f	['Denim is al	Fast Delivery	legitimate
1276	183f322706c	['"Zoom Vide	['Your payme	ca	legitimate
1277	183f017b5c9	['"Zoom Vide	['Zoom Invoi	As a	legitimate
1278	1838440908	['Zoom <no-i	['You are inv	Sarah has in	legitimate
1279	1833307423	['"Zoom Vide	['Payment Pr	Thank you	legitimate
1280	182fb3d50fe	['Instagram ·	['Verify your	Hi noname.4	legitimate
1281	182fb3bf88e	['Instagram ·	['Reset your	Hi noname.4	legitimate
1282	182f8b93fbb	['Harish <har	['Quote web	Hi dhai	legitimate
1283	182e899163	['"Zoom Vide	['Payment Pr	Thank you	legitimate
1284	18254cdbf7d	['Instagram ·	['Your Instag	This is a conf	legitimate
1285	18250eb7c6	['Team Snap	['New Snapc	Just logged i	legitimate
1286	18250eb496	['Team Snap	['Snapchat L	Snapchat Log	legitimate
1287	18213ca9e3	['no-reply@s	['[Action Rec	Your Jira site	legitimate
1288	<000001c6f5	"Comhghall I	Re: VdlAGRA	boundary="-	phishing
1289	<001501c71	fredrika tabb	My dear,	boundary="-	phishing
1290	<02e8710e-(	"HENRI MOY	INVESTMEN	boundary="a	phishing
1291	<0481ac3ea	"alice lacson	Dear partner	boundary="-	phishing
1292	<0HIG006X6	"Pandindra"	re:confidenti	Strictly Confi	phishing
1293	<0HPN006ZP	her29@redif	SINGAPORE	(iPlanet Me	phishing
1294	<0MKuxu-1C	"MR ABATO'	RESPONSE	MR ABATO (	phishing
1295	<0MKuxu-1C	"DONALD GA	SWIFT RESP	FROM=3ATH	phishing
1296	<0MKuxu-1C	"MR S. CHAR	RE: ACT AS T	Dear Partner	phishing
1297	<0MKuxu-1D	"Mrs.Erick" <	Confirm rece	mrsmary25@	phishing
1298	<0MKuxu-1D	"MR.ABBAH	GREETINGS	Hello my dea	phishing
1299	<0MKuxu-1D	" " <jh522g@	letter for bus	<html><heac	phishing
1300	<0MKuxu-1D	"Mrs.Angelir	God Bless Yc	Hello My Dea	phishing

Figure 26: Final data sample in 5.4.2

Due to some issues that CyberPhish faced with the implementation of AI, the dataset had to be modified. The original dataset, which was collected by the CyberPhish team and discussed in this section, was divided into two parts: legitimate and phishing. The legitimate

section was fully utilized as it was of high quality, while the phishing section was set aside. Subsequently, the CyberPhish team collected phishing data to complement the legitimate section of the dataset.

To create a more diverse and realistic email inbox, the team tried each of the three datasets (i.e., the original, spam and ham, and Enron datasets) in combination with the legitimate section to identify the best-fitting dataset. Spam and Ham and Ernon datasets, were used after removing the sender attribute, and using APIVoid to analyze the sender's email reputation. APIVoid provides JSON APIs useful for cyber threat analysis. This process resulted in a dataset containing 4742 tuples that was optimal. It had diversity, making it more realistic, and was unbalanced, mimicking a real-life inbox.

Creating a realistic and diverse dataset is crucial when it comes to cybersecurity solutions as suggested by Dr. Mohammad Almukaynizi during an AI bootcamp that CyberPhish attended. Mimicking a real-life inbox is one of the challenges that cybersecurity solutions face when it comes to data. Therefore, having a dataset that is diverse and unbalanced can help improve cybersecurity solutions by making them more robust and effective.

Having a diverse and unbalanced dataset is essential for training machine learning models used in cybersecurity. A more realistic and diverse dataset can help identify new and emerging threats and better prepare cybersecurity systems to defend against them.

Moreover, creating a diverse dataset can help address the issue of bias in machine learning models. When a dataset is biased, it can lead to biased results and inaccurate predictions. By creating a diverse and unbalanced dataset, the CyberPhish team was able to train their machine learning models to identify different phishing attacks.

4.5 Interface Design

In this section we will showcase the user interface design of CyberPhish, through the use of a user flow diagram. We will also discuss 7 UX guidelines that were implemented in CyberPhish.

4.5.1 User Navigation Diagram

In the user navigation diagram shown in Figure 27, we showcase the single point of access in CyberPhish, to model objects and their related screens. As well as the navigation paths available in CyberPhish.

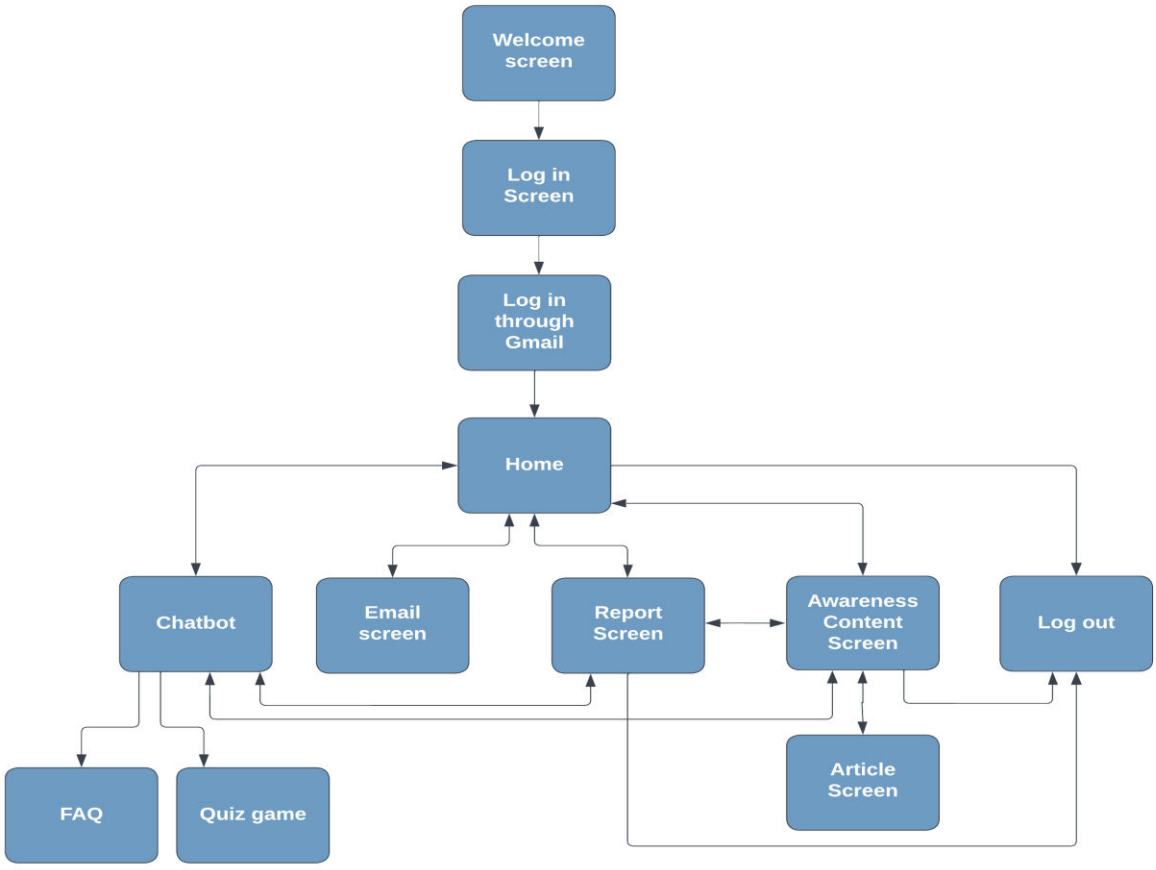


Figure 27: User Navigation Diagram in 5.5.1

4.5.2 UX Guidelines

- Learnability

For the learnability principle, CyberPhish followed four guidelines: consistency, generalizability, predictability, and familiarity, that are helping to make the CyberPhish application easier to use.

1. For consistency, there is consistency in the email inbox layout, displaying each email in a card as seen in Figure 28.

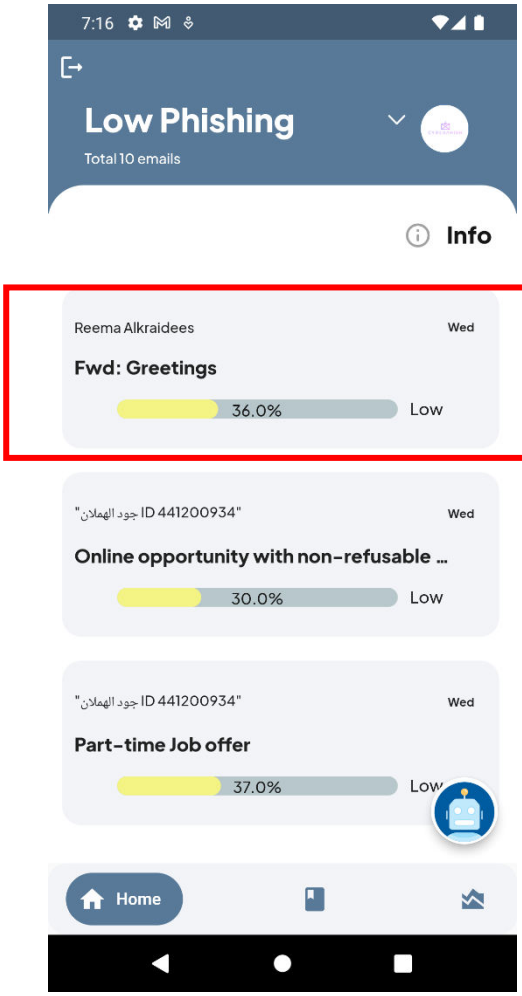


Figure 28: Home Screen in 4.5.2.1

2. For generalizability, the email inbox follows a card display layout similar to other email applications as shown in Figure 29.

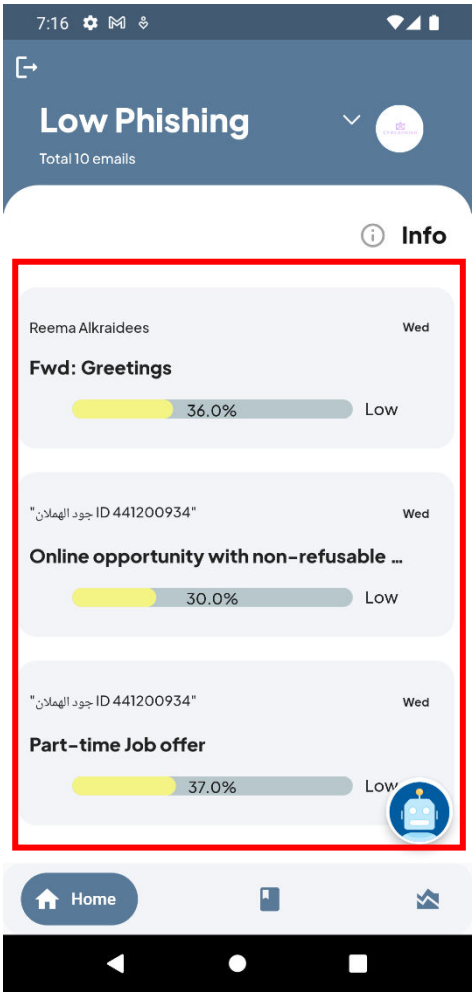


Figure 29: Home Screen in 4.5.2.1

3. For predictability, the user can predict that the ‘Log in using Gmail’ button will let them login through google as depicted in Figure 30.

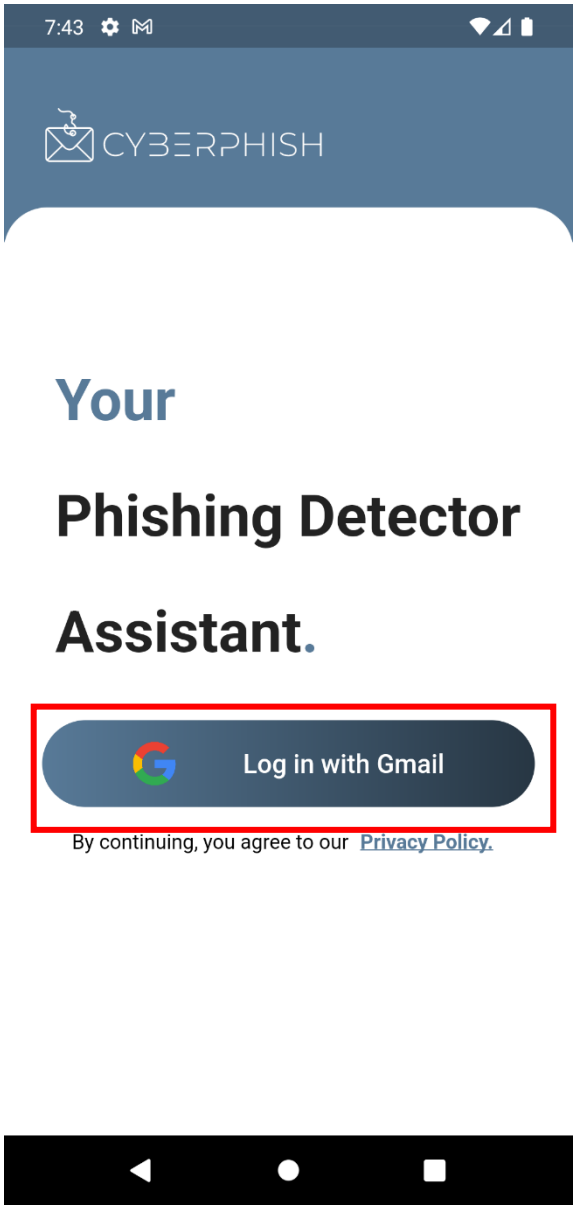


Figure 30: Log in Screen in 4.5.2.1

4. For familiarity, the ‘Log out’ button has a door icon which is familiar to the user, using it to go out as displayed in Figure 31.

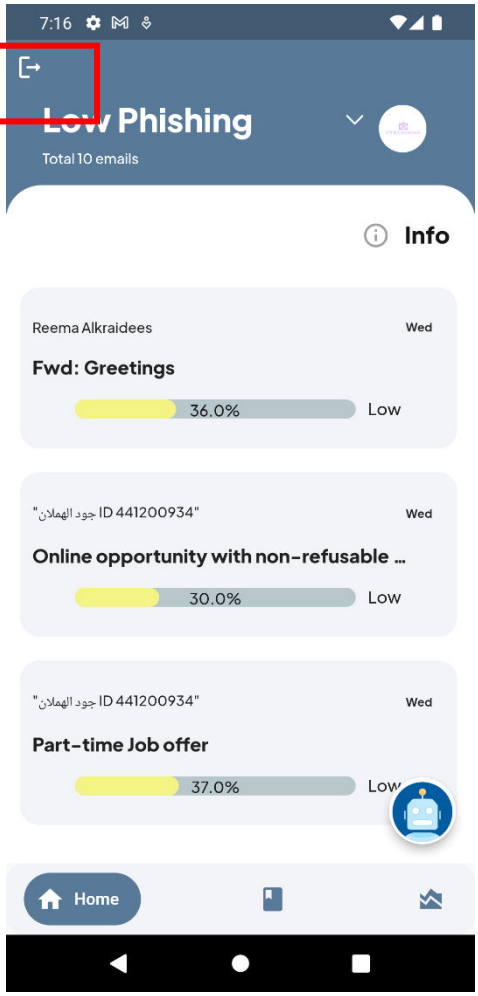


Figure 31: Home Screen-Logout Button in 4.5.2.1

- Flexibility

For the flexibility principle, CyberPhish followed the dialog initiative guideline, this guideline helps the user perceive the first action from the system which enhance the usability.

1. The system initiates the dialog in the chatbot when tapped as illustrated in Figure 32.

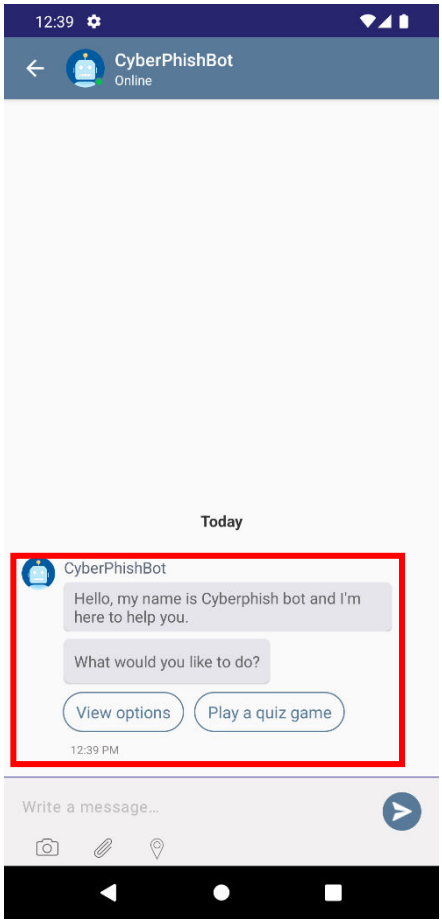


Figure 32: Chatbot Screen in 4.5.2.2

- Robustness

For the robustness principle, CyberPhish followed two guidelines: recoverability and reachability, these guidelines improve the robustness of CyberPhish application, therefore, the user can use the application and have full user experience with minimal errors.

1. For the recoverability, CyberPhish used buttons instead of free form user input in the chatbot as much as possible to avoid user errors as seen in Figure 33.

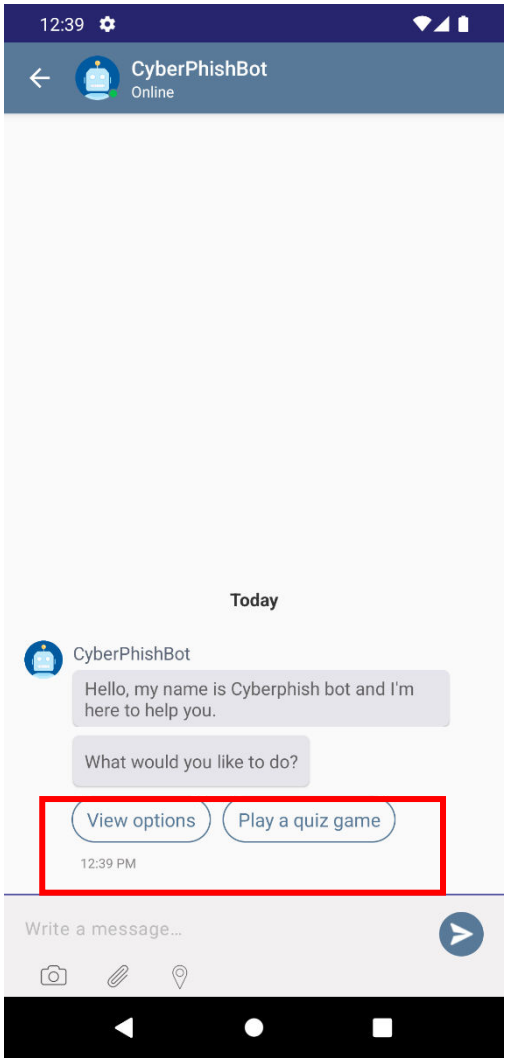


Figure 33: Chatbot Screen in 4.5.2.3

2. For reachability, in CyberPhish, the bottom navigation bar takes the user to their desired page as displayed in Figure 34.

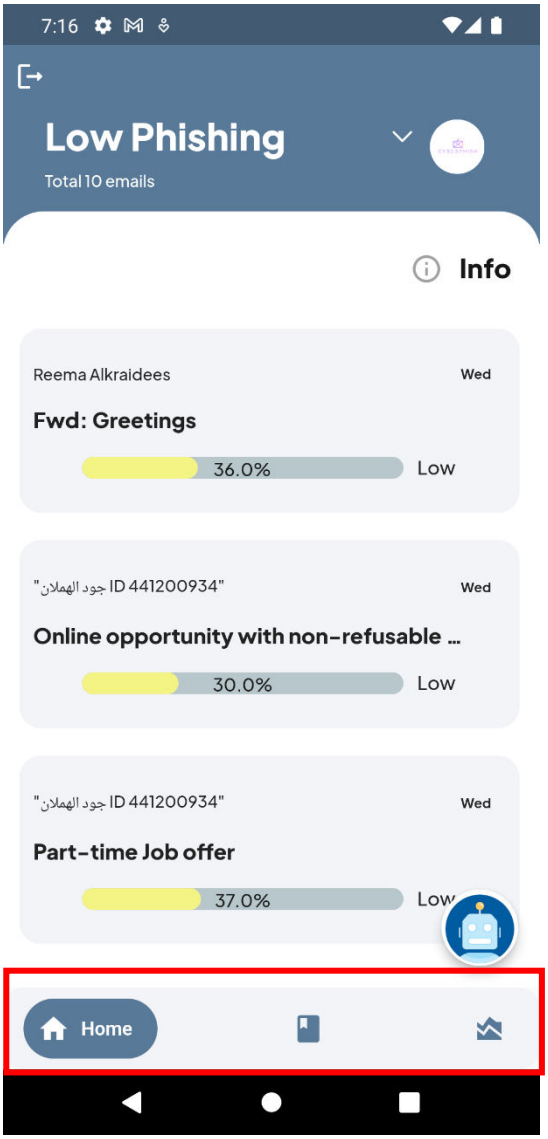


Figure 34:Bottom Navigation Bar in 4.5.2.3

4.6 Implementation

4.6.1 Login and Access to User Account

CyberPhish started with the user login process provided by the Google Gmail API, which grants CyberPhish access to the user's inbox and extracts emails. It then returns a JSON response that includes the Google sign in account object as seen in Figure 35, which

has the user's email, display name, id, and URL of the avatar, this will illustrate the user's account data in the firestore database as shown in Figure 36.

```
// instance of GoogleSignIn that allows us to use google sign in and sign out
  GoogleSignIn googleSignIn = GoogleSignIn(
    scopes: <String>[
      'email',
      'https://mail.google.com/', // scope that has full access
    ], // <String>[]
  ); // GoogleSignIn
```

Figure 35: Google sign in account object in 4.6

```
    // retrieve user's google account, store user's account in DB in user collection
    currentUser = await googleSignIn.signIn();
    await firestore
        .collection("GoogleSignInAccount")
        .doc(currentUser!.id)
        .set({
      "displayName": currentUser!.displayName,
      'email': currentUser!.email,
      'userId': currentUser!.id,
      'photoUrl': currentUser!.photoUrl,
      'userStatus': true,
    });
```

Figure 36: User's account data in 4.6

As the user logs in, the report data gets initialized by the current year, current month, and current week and each time frame takes the values for the triggersMap and senderMap and the two counter attributes for legitimate and phishing.

By sending a get profile request to Gmail API, the last 100 email IDs only will be returned as a JSON response, which is the maximum allowed limit by Google. Moreover, each email ID will be sent to the ExtractEmail function to extract the email's data as seen in Figure 37.

```dart
// send get request to get the last 100 emails from user inbox
final http.Response getProfile = await http.get(
  Uri.parse(
      'https://gmail.googleapis.com/gmail/v1/users/${user.id}/messages/'),
  headers: await user.authHeaders,
);
if (getProfile.statusCode != 200) {
  responseAPI = 'Gmail API  a ${getProfile.statusCode} '
      'response. Check logs for details.';
  debugPrint(responseAPI);
  return;
} else {
  // In successful getProfile request state:
  // Decode the json response, contain the last 100 email id
  final Map<String, dynamic> allEmailsResponse =
      json.decode(getProfile.body) as Map<String, dynamic>;
  emailIDlist = allEmailsResponse['messages'];

  // Loop on the emails, count how many recieved emails, to catch if there is less than 100 email
  for (var email in emailIDlist) {
    count++;
  }
  // Loop on to get the emails data, increment the check to prevent redundent request to users' inbox
  for (var i = 0; i < count; i++) {
    var emailId = allEmailsResponse['messages'][i]['id'].toString();
    emailCheck++;
    await Extractemail(emailId, user, emailsList).extractEmail(false);
  }
```

Figure 37: Get profile request to Gmail API in 4.6

4.6.2 Extracting Email

In "extractEmail" page, the extractEmail, extractHeader, extractDate, extractBody functions will work together to extract all the email data. Starting with extractEmail which is the primary function that retrieves the data from Gmail API using the emailData request and its JSON response as seen in Figure 38.

```dart
// send  get request to retrive specific email data using the email id
final http.Response emailData = await http.get(
  Uri.parse(
      'https://gmail.googleapis.com/gmail/v1/users/${user.id}/messages/$emailId'),
  headers: await user.authHeaders,
);

//decode the response, the response has all single email data, as 7 nested array, 100 fields of data
final Map<String, dynamic> emailDataResponse =
    json.decode(emailData.body) as Map<String, dynamic>;

// extract the headers contains all the header info such as sender, date, day, subject using loop through it
headers = emailDataResponse['payload']['headers'];
```

Figure 38: Email data request in 4.6

The extractEmail function is the primary function that calls many other functions, the first being extractHeader to extract date, sender email address, and name. The extractHeader calls another function called extractDate to extract the email date. The second function called by the extractEmail is extractBody which extracts the body data as seen in Figure 39.

```
// Map has the header data: subject, sender name, sender email, date, labels
emailHeaderMap = await extractHeader(headers, emailDataResponse);
// Map has the body data
bodyDataMap =
    await extractBody(emailDataResponse, emailHeaderMap['subject']);
// Map has the date data
dateMap = emailHeaderMap['mapDate'];
```

Figure 39: extractBody function in 4.6

The extractBody function takes a parameter which is emailDataResponse and will start to extract the body data by looping through it. The extracted body has various types, such as plain text, web page as HTML, and image attachments as seen in Figure 40.

```
// Loop on the email response recieved from Gamil
emailDataResponse.forEach((keyLayer1, valueLayer1) {
  // layer1
  // in Payload has the body and its parts
  if (keyLayer1.toString() == 'payload') {
    try {
      // multi parts emails
      if (valueLayer1['mimeType'].toString().contains('multipart')) {…
      } else if (valueLayer1['mimeType'] == 'text/plain') {
        body = utf8.decode(base64.decode(valueLayer1['body']['data']));
        // Extract HTML body and parse it
      } else if (valueLayer1['mimeType'].toString() == 'text/html') {
        body = utf8.decode(base64.decode(valueLayer1['body']['data']));
        bodyList['${++count}html'] = body;
      }
    } catch (e) {
      debugPrint('Error $e');
    }
  } // Payload closing
}); // email response loop closing
```

Figure 40: emailDataResponse function in 4.6

Then the body will be decoded, and parsed to split the text from the links, and remove any noisy data such as HTML tags. Only the body text will be sent to the predict function, in order to classify whether it is a phishing attempt or not using the CyberPhish AI model that has been uploaded to the server as illustrated in Figure 41.

```
// Start Parsing, split body from Links, claculate links' risk score, and classify body text
// will be used as the remaining not parsed yet body
remainBody = body.toString();
body = '';
while (remainBody != '') {
  String partParsedBody = '';

  // if remainBody has link
  if (remainBody.contains(RegExp("(http|https)?://"))) {
    try {
      var startIndex = -1, endIndex = -1;
      String partBody, partLink;

      // first index of link
      startIndex = remainBody.indexOf(RegExp("(http|https)?://"));
      if (remainBody.contains(linksExp, startIndex) && startIndex != -1) {
        // Last index of link
        endIndex = remainBody.indexOf(linksExp, startIndex);
        // substring the body part
        partBody = remainBody.substring(0, startIndex);
        // parsing body from HTML
        var doc = parse(partBody);
        if (doc.documentElement != null) {
          partParsedBody = doc.documentElement!.text;
        }
        // remove CSS and HTML tag
        if (partParsedBody.contains('>')) {
          partParsedBody =
              partParsedBody.substring(partParsedBody.indexOf('>') + 1);
        }
        // remove CSS and HTML tag
        if (partParsedBody.contains('}')) {
          partParsedBody =
              partParsedBody.substring(partParsedBody.indexOf('}') + 1);
        }
        partParsedBody = partParsedBody.replaceAll('<', ' ');
        partParsedBody = partParsedBody.replaceAll('>', ' ');
        partParsedBody = partParsedBody.replaceAll(' ', '  ');
        partParsedBody = partParsedBody.replaceAll('href="', ' ');
```

Figure 41: Decoding and parsing of body in 4.6

The response from the server includes the classification of whether the email is phishing or legitimate, and the vocabulary list which includes all the phishy words in the email as seen in Figure 42. As for the links they will be sent to checkURL function to check if it is a risky link or not using APIVoid as seen in Figure 43.

```
// Classify the text if phishy or legitimate and what are the vocabulary triggers
modelResponse =
    await APIviewmodel(user, emailsList).predict(subject, textBody);
vocabularyString = modelResponse['vocabulary'];
modelPrediction = double.parse(modelResponse['prediction']);
```

Figure 42: Response from the server in 4.6

```
// substring the link
partLink = remainBody.substring(startIndex, endIndex);
partLink = partLink.replaceAll(' ', '  ');
partLink = partLink.replaceAll('"', '  ');
// Check the URL risk score
await APIviewmodel(user, emailsList)
    .checkUrl(partLink, emailId, linkNum);
linkNum++;
```

Figure 43: checkURL function in 4.6

In order to calculate the percentage that will be displayed with each email, the checkSender and calculatePercentage functions will be used. The checkSender function will check the sender's email reputation using APIVoid and returns the sender's fraud score as seen in Figure 44.

```
final http.Response senderAPI = await http.get(Uri.parse(
    "https://endpoint.apivoid.com/emailverify/v1/pay-as-you-go/?key=$key&email=$senderEmail"));
final Map senderAPIResponse =
    json.decode(senderAPI.body) as Map<dynamic, dynamic>;
senderFraudScore = senderAPIResponse['data']['score'];
senderFraudScore = (senderFraudScore - 100) * -1;
```

Figure 44: Sender's email reputation using APIVoid in 4.6

The calculate percentage takes into account if the email has a link or not. The calculations were based on OWASP [36], which is a globally recognized authority on application security. OWASP identifies several common triggers that users should be aware of, including suspicious links, unusual senders, and the wording of the email itself, such as urgency, requests for sensitive information, and so on. Moreover, in a recent episode of Seen by Ahmad Al Shugairi, they highlighted the importance of link awareness and how over 110 billion attacks where because of clicks on malicious links [37].

If the email has a link, it takes the link risk score, sender's fraud score, the model's prediction, the words' weight, and the phishy words count. Each parameter was assigned a different weight. The link risk score has a 45% weight, sender's fraud score has a 15% weight, the model's prediction has a 15% weight, words' weight 25% weight as seen in figure 45. According to the final score, all phishy emails with links can be categorized into low, moderate or high. If the final score is 25 or below it is categorized as legitimate, if it is greater than 25 and less of equal to 50 it is categorized as low, if the final score is greater than 50 and equal to or less than 75 it is categorized as moderate, and lastly if the final score is above 75 it is categorized as high as seen in Figure 45.

```
// Email has links
if (linkNum >= 1) {
  // Get the max risk score of email's links
  await firestore
      .collection('GoogleSignInAccount')
      .doc(user.id)
      .collection("emailsList")
      .doc(emailId)
      .collection('links')
      .orderBy('RiskScore', descending: true)
      .limit(1)
      .get()
      .then(…
  riskScore ??= 0;
  // Percentage equation with links: Model Prediction 15%, Words 25%, sender risk score 15%, link risk score 45%
  percentage = (prediction * 15) +
      ((totalWeightWords * phisyWord * wordsPropotion) * 0.25) +
      (senderScore * 0.15) +
      (riskScore * 0.45);

  // classifying email category
  if (percentage <= 25) {
    predictionCategory = 'Legitmate';
  } else if (percentage > 25 && percentage <= 50) {
    predictionCategory = 'Low';
  } else if (percentage > 50 && percentage < 75) {
    predictionCategory = 'Moderate';
  } else if (percentage >= 75) {
    predictionCategory = 'High';
  }
```

Figure 45: Calculate percentage with link function in 4.6

If the email has no link the calculation is different. It takes into consideration the sender's fraud score, the model's prediction, the words' weight, and the phishy words count. Each parameter was assigned a different weight. The sender's fraud score has a 40% weight, the model's prediction has a 30% weight, words' weight 30% weight as seen in Figure 46. This weight scheme is based on the idea that if the email is without a clickable link the user will be less likely to accidently click on a malicious link; therefore, the chance of them falling for the phishing scam is less. According to the final score, all phishy emails with links can be categorized into low, moderate or high. If the final score is 30 or below it is categorized as legitimate, if it is greater than 30 and less of equal to 50 it is categorized as low, if the final score is greater than 50 and equal to or less than 75 it is categorized as moderate, and lastly if the final score is above 75 it is categorized as high as seen in Figure 46.

```
// Percentage equation without links: Model Prediction 30%, Words 30%, sender risk score 40%
percentage = (prediction * 30) +
    ((totalWeightWords * phisyWord * wordsPropotion) * 0.3) +
    (senderScore * 0.4);

// classifying email category
if (percentage <= 30) {
  predictionCategory = 'Legitmate';
} else if (percentage > 30 && percentage <= 50) {
  predictionCategory = 'Low';
} else if (percentage > 50 && percentage < 75) {
  predictionCategory = 'Moderate';
} else if (percentage >= 75) {
  predictionCategory = 'High';
}
}
```

Figure 46: Calculate percentage without link function in 4.6

After the categorization is done, the most trigger aspect of each phishing email is compared to find the trigger with the max effect on that specific phishing email as seen in Figure 47. All those data retrieved, extracted, parsed, calculated, and categorized will be saved in the Firestore database to be used later in other functions of CyberPhish.

```
// Finding the max trigger for an email
trigger = 'language';
maxtrigger =
    (prediction * 15) + ((totalWeightWords * phisyWord * wordsPropotion));
if (senderScore > maxtrigger) {
  maxtrigger = senderScore;
  trigger = 'sender';
} else if (riskScore > maxtrigger) {
  maxtrigger = riskScore;
  trigger = 'link';
}
```

Figure 47: Trigger of each phishing email in 4.6

4.6.3 Syncing

After extracting the last 100 emails, the stream function will take place. The stream is for monitoring and syncing any new changes in the user's inbox, such as a new email received, or an email deleted. This would be done using the history request, history response, history ID, and user status to check if the user has logged out or not as seen in Figure 48. If there are any new email updates they will be extracted as mentioned in the extraction steps before. As for deleted emails, they will also get deleted from the CyberPhish firestore database as seen in Figure 49. The history ID will be updated regularly as new changes come in.

```dart
// send post watch request, to monitor any changes on user Gmail account
final http.Response watchRequest = await http.post(
    Uri.parse(
        'https://gmail.googleapis.com/gmail/v1/users/${user.id}/watch'),
    headers: await user.authHeaders,
    body: jsonEncode(<String, String>{
      "topicName": 'projects/cyberphish-gp/topics/cyberphish'
    }));

// Decode the json response, contain the history id
final Map<String, dynamic> watchResponse =
    json.decode(watchRequest.body) as Map<String, dynamic>;

// Extract the history id, to use as parameter in the history request query parameter
historyId = watchResponse['historyId'];
userdata =
    await firestore.collection("GoogleSignInAccount").doc(user.id).get();
userStatus = userdata.data()!['userStatus'];

while (userStatus != false) {
  // User status flag to check if user logged out
  userdata =
      await firestore.collection("GoogleSignInAccount").doc(user.id).get();
  userStatus = userdata.data()!['userStatus'];

  // send get request to get any new changes and updates on the Gmail account
  final http.Response historyRequest = await http.get(
    Uri.parse(
        'https://gmail.googleapis.com/gmail/v1/users/${user.id}/history?startHistoryId=$historyId'),
    headers: await user.authHeaders,
  );

  // decode the json response, contain the new changes
  final Map<String, dynamic> historyResponse =
      json.decode(historyRequest.body) as Map<String, dynamic>;
```

Figure 48: New changes in the user's inbox in 4.6

```
// Extract the new change type and update based on it
try {
  //  history response map contain the new changes
  historyMap = historyResponse['history'];
  if (historyMap != null) {
    // Loop through each new change type update
    historyMap.forEach((key) {
      try {
        key.forEach((updateKey, updateValue) {
          // A new message recieved
          if (updateKey == 'messagesAdded') {
            updateValue.forEach((messageUpdate) {…
          }
          // message deleted
          if (updateKey == 'messagesDeleted') {
            updateValue.forEach((messageUpdate) {…
          }
        });
      } catch (e) {
        debugPrint('error in second $e');
      }
      try {
        historyId = key['id'];
      } catch (e) {
        debugPrint('Error in history $e');
      }
    });
  } else {
    // No new change, only update the history Id
    historyId = historyResponse['historyId'];
  }
```

Figure 49: Deleted email in CyberPhish datastore in 4.6

4.6.4 Display home

The home screen is the default page in CyberPhish after the user logs in. This page retrieves the user's inbox data which has all the email data including emailId, subject, senderName, senderEmail, body, bodyList, date, percentage, senderFraudScore, linkRisk, prediction, wordsPropotion, wordsList, and valuesList. Then, save these data as an email object as seen in Figure 50. The email object will be sent to the mail card and email screen pages.

```
child: StreamBuilder<QuerySnapshot>(
  stream: firestore
      .collection("GoogleSignInAccount")
      .doc(widget.user.id)
      .collection("emailsList")
      .snapshots(),
  builder: (context, snapshot) {
    List<MailCard> mailcardList = [];
    if (!snapshot.hasData) {…
    }
    final mails = snapshot.data!.docs.reversed;
    for (var mail in mails) {
      final emailId = mail.get('emailId');
      final subject = mail.get('subject');
      final senderName = mail.get('senderName');
      final senderEmail = mail.get('senderEmail');
      final body = mail.get('body');
      final date = mail.get('date');
      final prediction = mail.get('prediction');
      final percentage = mail.get('percentage').toString();
      final Map bodyList = mail.get('bodyList');
      final List wordsList = mail.get('wordsList');
      final List valuesList = mail.get('valuesList');
      final double wordsPropotion = mail.get("wordsPropotion");
      var linkRisk = mail.get('linkRisk');
      var senderFraudScore = mail.get('senderRiskScore');

      try {
        firestore
            .collection('GoogleSignInAccount')
            .doc(widget.user.id)
            .collection("senders")
            .where('email', isEqualTo: senderEmail)
            .limit(1)
            .get();
      } catch (e) {
        debugPrint('Error $e');
      }
      widget.emailsList.add(
        // add an email to the list, using the email class
        Email( // Email …
      );
      final mailwidget = MailCard(
        email: Email( // Email …
        user: widget.user,
      ); // MailCard
```

Figure 50: Email object in 4.6

The inbox is constructed using mail cards, where each mail card has the email object data as seen in Figure 51. The information provided to the user with each mail card is the

sender's name, sender's email, subject, date, flag, and percentage if the email was marked as phishing.

```
child: ListView(
  children: mailcardList,
  reverse: false,
  padding: const EdgeInsets.only(top: 16),
), // ListView
```

Figure 51: Mail cards function in 4.6

If the user chooses a specific email to read, they will be directed to the email screen page. The email screen page receives the email object and displays the data on the screen. The email header has the subject, sender email, and the date. The email body is displayed based on its type using the bodyBuilder function, which puts each type separately and in its usual form, whether it is text, web, or image attachment as shown in Figure 52.

```
if (keyList.toString().contains('html')) {
  // has html
  if (indexKey.toString().contains('html')) {…
  }
} else {
  // no html
  if (indexKey.toString().contains('link')) {
    var newString = part!.substring(part!.length - 5);
    if (newString.contains('png') ||
        newString.contains('jpg') ||
        newString.contains('jpeg') ||
        newString.contains('gif')) {…
```

Figure 52: bodybuilder function in 4.6

4.6.5 Display report

The report page starts with initializing the legitimate counter, phishing counter, senderEmail, and triggersMap, and give those counters an initial value of zero or null values according to the counter's type as seen in Figure 53, which would be used in the charts of the report page. These values are stored based on their real lifetime in days, weeks, months, and year as seen in Figure 53.

```
await firestore
    .collection("GoogleSignInAccount")
    .doc(currentUser!.id)
    .collection("report")
    .doc('${DateTime.now().year}')
    .set({
  'totalY${DateTime.now().year}': {
    'legitmate': 0,
    'phishing': 0,
    'senderEmail': '',
    'triggersMap': {
      'language': 0,
      'sender': 0,
      'link': 0,
    },
  },
  '${DateTime.now().month}': {
    'totalM${DateTime.now().month}': {…
    'w$weeknum': {
      '${DateTime.now().day}': {…
      'totalW$weeknum': {…
```

Figure 53: initialization of counters in 4.6

The linear chart displays the total number of phishing emails and the total number of legitimate emails; therefore, we used the two counters. Firstly, the legitimate counter which gets incremented each time a new legitimate email is extracted from the user's inbox as seen in Figure 54. Secondly, the phishing counter which gets incremented each time a new phishing email is extracted from the user's inbox as seen in Figure 55.

```
await firestore
    .collection("GoogleSignInAccount")
    .doc(user.id)
    .collection("report")
    .doc('${dateMap['year']}')
    .update({
  '${dateMap['month']}.w${dateMap['week']}.${dateMap['dayNumber']}.legitmate':
      FieldValue.increment(1),
  '${dateMap['month']}.w${dateMap['week']}.totalW${dateMap['week']}.legitmate':
      FieldValue.increment(1),
  '${dateMap['month']}.totalM${dateMap['month']}.legitmate':
      FieldValue.increment(1),
  'totalY${dateMap['year']}.legitmate': FieldValue.increment(1),
```

Figure 54: Increment legitimate counter in 4.6

```
await firestore
    .collection("GoogleSignInAccount")
    .doc(user.id)
    .collection("report")
    .doc('${dateMap['year']}')
    .update({
  '${dateMap['month']}.w${dateMap['week']}.${dateMap['dayNumber']}.phishing':
      FieldValue.increment(1),
  '${dateMap['month']}.w${dateMap['week']}.totalW${dateMap['week']}.phishing':
      FieldValue.increment(1),
  '${dateMap['month']}.totalM${dateMap['month']}.phishing':
      FieldValue.increment(1),
  'totalY${dateMap['year']}.phishing': FieldValue.increment(1),
```

Figure 55: Increment phishing counter in 4.6

The insights aim to summarize the user's inbox regarding the total number of phishing emails and the most dangerous sender in a chosen timeframe. The total number of phishing emails has the phishing email count using the phishing counter as shown in Figure 55. As for the most dangerous sender, it displays the highest-ranking sender email which has the highest risk score to the user, using a comparison technique as seen in Figure 56. Each time the highest-ranking sender email changes, the value is updated in firebase.

```
if (keyYear.toString() == ('totalY${dateMap['year']}')) {
  // find Total Year
  if (senderRiskScore > valueYear['senderEmail']) {
    newEmailYear = emailHeaderMap['senderEmail'];
  }
  if (senderRiskScore <= valueYear['senderEmail']) {
    newEmailYear = valueYear['senderEmail'];
  }
```

Figure 56: Most dangerous sender in 4.6

The pie chart aims to display all triggers in the entire user's inbox with their corresponding values by using the triggersMap. The triggersMap has the language counter, sender counter, and link counter, which highlight the single email's most triggering aspect among those three counters, as seen in Figure 57. The trigger variable takes the highest value, which is then incremented in firebase, as seen in Figure 58.

```
// Finding the max trigger for an email
trigger = 'language';
maxtrigger =
    (prediction * 15) + ((totalWeightWords * phisyWord * wordsPropotion));
if (senderScore > maxtrigger) {
  maxtrigger = senderScore;
  trigger = 'sender';
} else if (riskScore > maxtrigger) {
  maxtrigger = riskScore;
  trigger = 'link';
}
```

Figure 57: triggersMap in 4.6

```
'${dateMap['month']}.w${dateMap['week']}.${dateMap['dayNumber']}.triggersMap.$trigger':
    FieldValue.increment(1),
'${dateMap['month']}.w${dateMap['week']}.totalW${dateMap['week']}.triggersMap.$trigger':
    FieldValue.increment(1),
'${dateMap['month']}.totalM${dateMap['month']}.triggersMap.$trigger':
    FieldValue.increment(1),
'totalY${dateMap['year']}.triggersMap.$trigger':
    FieldValue.increment(1),
```

Figure 58: Update maxrigger in 4.6

The report page data is retrieved all at once, then looping through the maps, each time frame has its own values. The year map has the total number of phishing emails, legitimate emails, triggersMap values, and senderEmail value of the current year as seen in 59. The same process is applied to the months, and weeks with the same maps but values of that specific time frame. As for the extracted data of each time frame, it will be sent to its own time frame page, to be displayed to the user based on their choice as illustrated in Figure 60.

```
child: StreamBuilder<QuerySnapshot>(
stream: firestore
        .collection("GoogleSignInAccount")
        .doc(widget.user.id)
        .collection("report")
        .snapshots(),
  builder: (context, snapshot) {
    if (!snapshot.hasData) {
      return const Center(child: Text('No Reports yet!'));
    }
    final reports = snapshot.data!.docs.toList();
    // retriving this year's data, and updating the initialized year related variables.
    int yearlyPhishing = 0;
    int yearlyLegitmate = 0;
    for (var year in reports) {
      yearlyPhishing = year.get('totalY${year.id}')['phishing'];
      yearlyLegitmate = year.get('totalY${year.id}')['legitmate'];
      yearlyTotalPhishy = yearlyPhishing.toDouble();
      yearlyTotalLegitmate = yearlyLegitmate.toDouble();
      yearlyTriggerMap =
          year.get('totalY${year.id}')['triggersMap'];
      yearlySenderEmail =
          year.get('totalY${year.id}')['senderEmail'];
      map = year.data() as Map?;
```

Figure 59: Year map in 4.6

```
// dashboard to be displayed is initialy this year's. then it changes acoording to users' choice.
Widget dashboard = YearDashboard(
    totalLegitmate: yearlyTotalLegitmate,
    totalphishy: yearlyTotalPhishy,
    yearlyLegitmateMap: yearlyLegitmateMap,
    yearlyphishingMap: yearlyphishingMap,
    yearlyTriggerMap: yearlyTriggerMap,
    yearlySenderEmail: yearlySenderEmail); // YearDashboard

// changing the dashboard to be displayed according to user's choice.
// sending the required data to that dashboard screen.
if (isSelected[0] == true) {
  dashboard = YearDashboard(
      totalLegitmate: yearlyTotalLegitmate,
      totalphishy: yearlyTotalPhishy,
      yearlyLegitmateMap: yearlyLegitmateMap,
      yearlyphishingMap: yearlyphishingMap,
      yearlyTriggerMap: yearlyTriggerMap,
      yearlySenderEmail: yearlySenderEmail); // YearDashboard
} else if (isSelected[1] == true) {
  dashboard = MonthDashboard(
    totalphishy: monthlyTotalPhishy,
    totalLegitmate: monthlyTotalLegitmate,
    monthlyLegitmateMap: monthlyLegitmateMap,
    monthlyphishingMap: monthlyphishingMap,
    monthlyTriggerMap: monthlyTriggerMap,
    monthlySenderEmail: monthlySenderEmail,
  ); // MonthDashboard
} else {
  dashboard = WeekDashboard(
    totalphishy: weeklyTotalPhishy,
    totalLegitmate: weeklyTotalLegitmate,
    weeklyLegitmateMap: weeklyLegitmateMap,
    weeklyphishingMap: weeklyphishingMap,
    weeklyTriggerMap: weeklyTriggerMap,
    weeklySenderEmail: weeklySenderEmail,
  ); // WeekDashboard
```

Figure 60: Extraction of report data in 4.6

4.6.1 Display awareness

The awareness page has many articles, with the aim to raise the awareness of the user regarding phishing. The awareness page's data retrieving begins when the application starts. The title, author, link, and image link attributes are added to the article object, as seen in Figure 61.

```
final articles = await firestore.collection('Awareness').get();
for (var articleData in articles.docs) {
  articleList.add(
    // add an article to the list, using the article class
    Article(
        title: articleData.data()['title'],
        author: articleData.data()['author'],
        link: articleData.data()['link'],
        imgLink: articleData.data()['imgLink']),
  );
}
```

Figure 61: Retrieve data of articles in 4.6

The awareness page has an article list which contains all the article cards. The article cards display an overview of the article's title, author, and image. The user is then navigated to the article screen when they click on the article card. In the article screen the article is displayed as a webpage using the article's links as illustrated in Figure 62.

```
child: WebView(
  initialUrl: "${article?.link}",
  javascriptMode: JavascriptMode.unrestricted,
  onProgress: (progress) => Text('in progress ... $progress'),
), // WebView
```

Figure 62: Article webpage in 4.6

4.6.2 Chatbot

In order to develop the chatbot, an account was established on Dialogflow. Initially, a set of frequently asked questions ("FAQs") was incorporated as phrases that the chatbot could utilize to provide responses. Subsequently, the aim was to enhance the user experience by integrating "suggested reply" buttons. To accomplish this, an account was created on Kommunicate and then proceeded to integrate the Dialogflow account with the Kommunicate account. Dialogflow and Kommunicate's integration was necessary to facilitate the addition of suggested replies to buttons. Moreover, a quiz game feature got incorporated within the chatbot, consisting of a predetermined sequence of ten questions to evaluate the user's knowledge regarding phishing. The chatbot is then integrated with the CyberPhish application using an API key called appId as visualized in Figure 63.

```
floatingActionButton: FloatingActionButton(
  onPressed: () async {
    try {
      dynamic conversationObject = {
        'appId': '3b1a0def678f8cccb9c44ea7dd5065d9f',
      };
      await KommunicateFlutterPlugin.buildConversation(conversationObject)
          .then((clientConversationId) {});
    } on Exception catch (e) {
      debugPrint('Error $e');
    }
  },
```

Figure 63: Chatbot integration in 4.6

4.6.3 Logout

When a user logs out from their account a request is sent to Google in order to stop the watch function, which keep monitoring the new changes on the user's inbox. The "userStatus" flag, that indicates if the user is still logged in or not, will be false and the emailStream function will be stopped. The logged-out user's GoogleSignInAccount is then deleted from the firestore database including all the emails, report data as demonstrated in Figure 64.

```
// handle sign out, empty the email list, return the user to log in screen
handleSignOut(GoogleSignInAccount currentUser) async {
  await http.post(
    Uri.parse(
        'https://gmail.googleapis.com/gmail/v1/users/${currentUser.id}/stop'),
  );
  firestore
      .collection("GoogleSignInAccount")
      .doc(currentUser.id)
      .set({'userStatus': false});
  Get.offAll(() => const LoginScreen());
  await firestore
      .collection("GoogleSignInAccount")
      .doc(currentUser.id)
      .collection("emailsList")
      .get()
      .then((querySnapshot) async {…

  firestore
      .collection("GoogleSignInAccount")
      .doc(currentUser.id)
      .collection("report")
      .doc('${DateTime.now().year}')
      .delete();

  await firestore
      .collection("GoogleSignInAccount")
      .doc(currentUser.id)
      .delete();
  emailsList = [];
  await googleSignIn.signOut();
}
```

Figure 64: Logout function in 4.6

4.6.4 API model

The CyberPhish SVM model and the CyberPhish vectorizer were converted to pickle (pkl) format to be integrated with the application. By saving a trained model using the pickle module, you can reuse the model for making predictions on new data, without having to retrain the model from scratch [31]. Then they were uploaded to the server, as illustrated in Figure 65. After receiving a request from the application containing the subject and the text body of the email, they will be integrated together to be a feature variable. The feature variable is then sent to the model with a predict function to get the classification label of the sent email. The vectorizer function is then used to get the bag of words and the vocabulary list of the emails. The vocabulary list has the most phishing words found in the email. Lastly, a JSON response will be returned, as seen in Figure 66.

```
model = pickle.load(open('SVMmodel.pkl', 'rb'))
vectorizer = pickle.load(open("vectorizer.pkl", "rb"))
```

Figure 65: Model and Vectorizer in 4.6

```
data = request.data
request_data = json.loads(data.decode('utf-8'))
subject = request_data['subject']
body = request_data['body']
features = "\n".join([subject, body])
prediction = model.predict([features])
encode  = vectorizer.transform([features]).toarray()
bag_of_words = pd.DataFrame(
        encode, columns=vectorizer.get_feature_names_out())
Vocab_list = {}
for vector in bag_of_words:
    if (bag_of_words[vector].values[0] > 0):
        Vocab_list[bag_of_words[vector].name] = bag_of_words[vector].values[0]
prediction = f'{prediction}'
response = f'{Vocab_list}'
return jsonify({'prediction': prediction[1],
                'vocabulary': response
                })
```

Figure 66: API model in 4.6

GitHub: https://github.com/Cyberphish/2022-GP1-G8

5 System Evaluation

In this section the AI models' experimental results and user acceptance testing results will be presented. The AI models were trained and evaluated in order to discover the model that garnered the best overall result, then it will be implemented within the application. tests were conducted on the completed CyberPhish application by the help of the application's potential users. These tests were performed with the aim of evaluating the system's compliance with its specified requirements.

5.1 Experimental Results

The model training and testing processes have been conducted on a software called Orange Data Mining. Orange is an open-source data visualization, machine learning and data mining toolkit. It features a visual programming front-end for explorative rapid qualitative data analysis and interactive data visualization [18]. For a speedier procedure, the comparison of the three models was performed using Orange Data Mining.

The general process began by importing the dataset into the software and then creating a table to visualize the data. After that, the dataset was split 70/30 using the "data sampler" widget; the 70 percent was used for training and the 30 percent for testing. The "corpus" widget was used on the training data, and then the text was preprocessed via the "preprocess text" widget. During the preprocessing procedure, the text was filtered; stop words and regular expressions were removed. Next, we selected the columns, similar to feature selection, where we selected the target to be the label (phishing or not), and the subject and body were the other columns. After that, we used the "bag of words" widget, that was then connected to the "learner" (the model) and the "test and score" widget.

The test and score widget allows to select a method of testing such as K-fold cross-validation or leave-one-out cross-validation. Based on our research, we opted to use 10-fold cross-validation, which is recognized to be an effective method for obtaining unbiased or nearly unbiased estimates of error rates for classification and prediction based on a given size of training set. Also, 10-fold cross-validation is the most commonly used by researchers working in the same domain as ours, and it gave us the best results [15].

5.1.1 Random Forest

The general processes were done for the Random Forest model, from the importation of the dataset to the same sequence of widgets all connected to the "test and score" and

"learner" widgets. 10-fold cross-validation was used for the "test and score" widget. With this model, there is an option to choose the number of trees generated; more trees give better results when training as shown in Table 4.

Table 4: Comparison between trees in section 5.1.1

Model	Area Under Curve	Classification Accuracy	F1 Score	Precision	Recall
Random Forest-30 tree	0.972	0.915	0.915	0.917	0.915
Random Forest-20 tree	0.972	0.915	0.914	0.915	0.915
Random Forest-10 tree	0.969	0.907	0.906	0.907	0.907
Random Forest 50 tree	0.973	0.916	0.916	0.918	0.916
Random Forest 200 tree	0.975	0.915	0.915	0.918	0.915
Random Forest 100 tree	0.974	0.916	0.916	0.918	0.916

The random forest model works by starting with a set number of trees, each of which performs its own classification process and result in its own prediction, and then the final prediction result of the single model is based on the average classification prediction result of its trees. We compared the results of the prediction of different number of trees, we tried 10 which is the minimum number of trees allowed. We increased the number of trees by trying 20, 30, 50, 100, and 200 trees.

The fundamental benefit of include a high number of decision trees in your random forest model is that prediction performance improves as the number of trees increases. It is important to mention that after a certain point, the results will reach a peak, and then it will reach a point of diminishing returns, where the scale of the performance improvements you observe from inserting additional trees will get progressively smaller. The result has reached its peak at 50 trees, and then as more trees are added, the result starts to decline due to the redundancy of trees as seen in Table 5.

Table 5: Results of model in section 5.1.1

Model	Area Under Curve	Classification Accuracy	F1 Score	Precision	Recall
Random Forest-30 t	0.969	0.897	0.896	0.904	0.897
Random Forest-20 t	0.968	0.892	0.892	0.901	0.892
Random Forest-10 t	0.964	0.885	0.885	0.896	0.885
Random Forest 50 t	0.970	0.901	0.901	0.908	0.901
Random Forest 100 t	0.970	0.899	0.899	0.906	0.899
Random Forest 200 t	0.970	0.898	0.898	0.905	0.898

5.1.2 SVM

In the SVM model, the general sequence of widgets was used: the "corpus", "text preprocessing", "selected columns", "bag of words", "test and score" and "learner" widgets.

The difference was that a "preprocessing" widget was added before the model to act as an empty preprocessing step as seen in figure 67, because SVM uses default preprocessing when no other preprocessors are given. We did the empty preprocessing since the default preprocessing options caused issues with our SVM model and kept saying the data was sparse. We reached this solution after doing some research on the way the SVM widget works in Orange Data Mining.

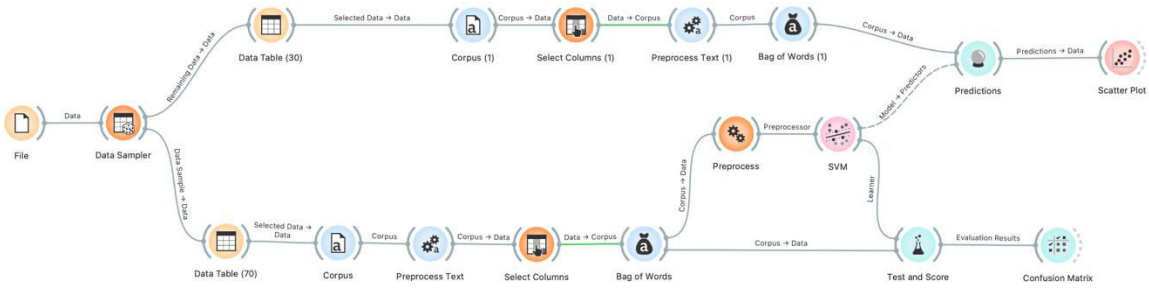


Figure 67: Orange DM for SVM in section 5.1.2

There are two types of SVM in Orange: SVM and v-SVM which are based on different minimization of the error function. Both SVM and v-SVM have the cost parameter, which is a penalty term for loss, in SVM it is used for both regression and classification tasks, while in v-SVM it is used for regression only. Due to that, the type we utilized was SVM, and the default cost value is 1 which we altered to 0.9 and that gave us a better result. We then tried 0.8 which gave an even better result than 0.9. For that reason, we tried to reduce the cost even more by trying 0.7 and 0.6, but the results decreased which terminated the reduction trials. 0.8 produced the best results as shown in Table 6 hence we choose it.

Table 6: Results of model in 5.1.2

Model	Cost	Area Under Curve	Classification Accuracy	F1 Score	Precision	Recall
SVM	1	0.92	0.67	0.63	0.79	0.67
SVM	0.90	0.91	0.67	0.63	0.79	0.67
SVM	0.80	0.91	0.67	0.63	0.79	0.67
SVM	0.70	0.92	0.66	0.62	0.79	0.66
SVM	0.60	0.91	0.64	0.59	0.78	0.64

The kernel that was chosen was a linear kernel because it is the most common, used in a variety of different ways, and used when data can be separated using a single Line [11]. After that we performed 10-Fold cross-validation, where we got the results and values for all the parameters.

Next, we did the same widget sequence on the 30% testing data, where the "bag of words" widget and the model were then connected to the "predictions" widget. After getting the results from the "prediction" widget, we visualized the data using a scatter plot as shown in figure 68, without the confusion matrix because it wasn't available for the SVM model.

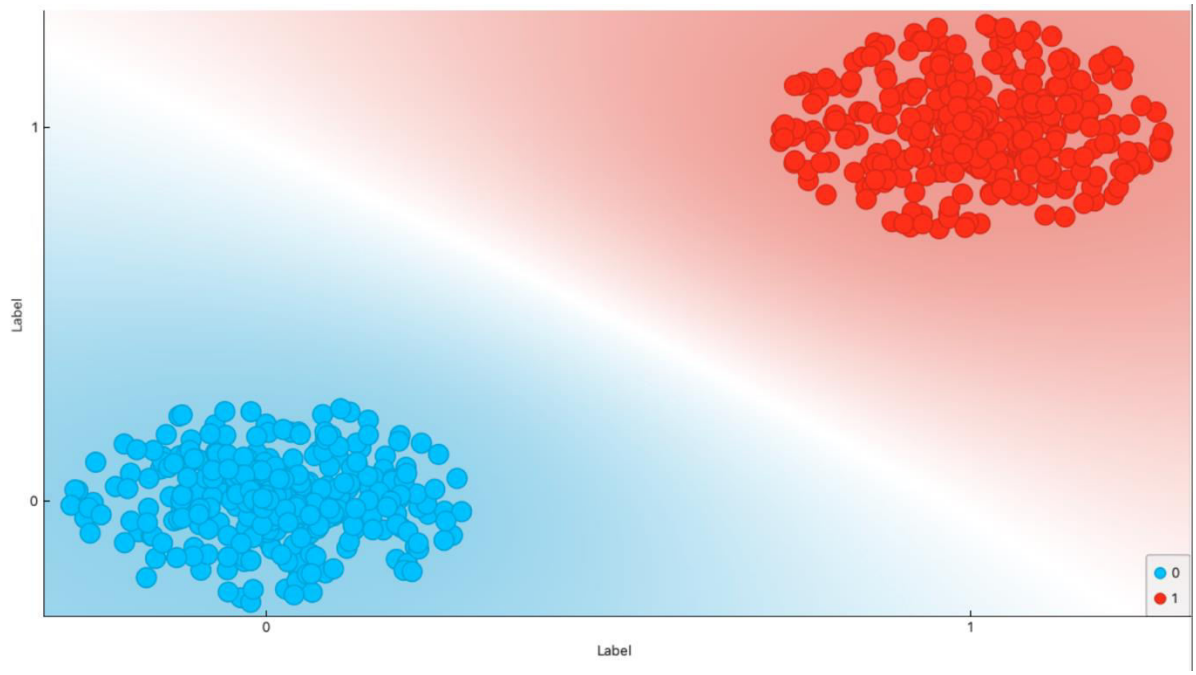


Figure 68: Scatterplot in 5.1.2

5.1.3 Naïve Bayes

For the Nave Bayes model, none of the parameters changed like the SVM model, so we saved the model's score that was given by the "Test and Score" widget. The model was then connected to the "Predictions" widget.

The 30% testing data was then subjected to the same procedures as the training data, where the corpus, text preprocessing, selected columns, and bag of words widgets were used in that order. Then we connected the bag of words widget to the model, where we got the predictions that resulted in a confusion matrix, data visualization using a scatter plot, and the ROC analysis.

5.1.4 Orange DM Troubleshooting

After training and testing all three models, Naïve Bayes was to be implemented on the CyberPhish application, but the model couldn't be imported from Orange Data Mining to

Python in order to integrate it with CyberPhish application, so we had to do the training and testing processes again from scratch using Python.

The first step was to import the dataset onto Python script. Next, we created a column named “Features” that contains the subject and body of the emails that were merged, so now there are two attributes: Label and Features. Following that, we developed a text preprocessing method to remove stop words similar to the one used by Orange Data Mining, in which stop words such as "I," "them," "what," and so on are removed. Following that, we created a word vectorizer to construct the bag of words.

At this stage, we split the dataset into 70/30, with 70 percent used for training and 30 percent for testing. This split was chosen in relevance to the size of our dataset as to have a good flow of testing data [32]. For the training and testing part, we had to do some research to find the best way to implement the Naïve Bayes model into the Python code. There were different algorithms of Naïve Bayes, but we chose the best one for our dataset, which is Complement Naïve Bayes, which we fitted onto the code. Afterwards, we started the prediction and visualization process by generating ROC curve, confusion matrix, and accuracy.

Next, 10-Fold-Cross-Validation was implemented and then a process called “pickle dump” was executed to convert the model into a pickle file so it can be used with the API and the application.

There were many modeling techniques tried and tested to get the optimum CyberPhish model, starting with the use of the Orange Data Mining Toolkit to decide which algorithm to use. CyberPhish implemented the three AI algorithms in Orange DM, where the Naïve Bayes algorithm showed the best performance out of the three. When the Naïve Bayes algorithm was then implemented on Python using the same steps used in Orange, the results were faulty. The model was tested on the API to ensure the integration was not an issue. Then the model got deployed to the server to be tested on a CyberPhish user's inbox.

The maximum accuracy reached by the deployed model was only in the range of 65 to 68. Then the trails moved on to the multicast Naïve Bayes Python code, where a pipeline method was not used, and the data was not unbalanced therefore, was not imitating real world user inboxes. There were more than 55 recorded trials to look for the combination of attributes that resulted in the highest accuracy as seen in Table 7. The results seemed to still

be poor, with a lower score of 67 or less on the API. All those efforts were put aside, and the three AI algorithms were written again using only python and the pipeline method.

Table 7: Trails in 5.1.4

Min_df	Max_df	# columns (features)	N-folds	Accuracy w/n-folds	Accuracy API	Seed
5	700	4474	10	96.427	62	1000
5	600	4589	10	95.873	67	1000
5	500	4485	10	96.284	57	1000
10	600	2675	10	96.006	66	1000
15	600	1933	10	96.564	63	1000
15	800	1959	10	96.016	64	1000
15	1000	1966	10	97.112	62	1000
20	800	1594	10	96.290	64	1000
25	700	1354	10	96.286	61	1000
25	500	1350	10	96.427	55	1000
50	900	711	10	96.693	48	1000
2	900	10264	10	94.500	60	1000
2	1000	10182	10	95.462	61	1000
2	2000	10074	10	93.674	64	1000
1	2000	27617	10	91.216	59	1000
2	3000	10095	10	94.770	60	1000
2	2500	10086	10	94.368	62	1000
2	2400	10061	10	94.502	63	1000

5.1.5 CyberPhish's implementation of SVM

CyberPhish's journey with Python for AI implementation began with the dataset. There were four types of datasets tested; the first dataset was the original dataset that was collected by the CyberPhish team, as mentioned in section 4.4.2. The second dataset was spam and ham, which had both spam and ham and Enron incorporated together. The third dataset was an organization's dataset belonging only to Enron. The fourth dataset was a collection of the three previously mentioned datasets, from which a small part of each was taken.

To create a realistic email inbox dataset for cybersecurity solutions, we first divided the original dataset into two parts: legitimate and phishing. As the legitimate section was of high quality, we fully utilized it and set aside the phishing section. We then collected additional phishing data to complement the legitimate section, trying each of the three

datasets - original, spam and ham, and Enron - in combination to find the best-fitting one. After careful selection, we obtained a final dataset with 4742 tuples that was diverse and unbalanced, mirroring the real-life challenges faced by cybersecurity solutions. This optimal dataset provides a more realistic representation of a typical inbox. The word cloud of this dataset clearly showed the impact of the significant words without the noisy and unnecessary words, as shown in Figure 69.

Figure 69: Word Cloud in 5.1.6

Then the final dataset was taken and used to train the machine learning algorithm using Python. We commenced with the preprocessing first. Stop words were removed, all the characters were lowered for uniformity, and the contracted words were expanded into their uncontracted form (i.e., "aren't" becomes "are not") as seen in figure 70.

```
def expand(x):
    if type(x)== str:
        for key in contractions:
            value = contractions[key]
            x = x.replace(key,value)
        return x
    else:
        return x
```

Figure 70: Expand function in 5.1.5

Lemmatization was used to convert the words to their base word and dictionary head word. Accented characters were also changed to their equivalent normal form. Using regex,

all the punctuation got removed. Trailing spaces and extra spaces between words were removed as illustrated in figure 71.

```
def remove_accented_chars(x):
    x = unicodedata.normalize('NFKD', x).encode('ascii', 'ignore').decode('utf-8', 'ignore')
    return x

def make_to_base(x):
    x_list = []
    doc = nlp(x)

    for token in doc:
        lemma = str(token.lemma_)
        if lemma == '-PRON-' or lemma == 'be':
            lemma = token.text
        x_list.append(lemma)
    return (" ".join(x_list))


def preprocess(df,d):
    df[d] = df[d].apply(lambda x: " ".join(x.split()))
```

Figure 71: lemmatization and accented characters functions in 5.1.5

Lastly, polarity was taken into account, which is the sentiment of the word, from -1(negative) to 1(positive) in a given sentence as seen in figure 72 of the emails' polarity collected by CyberPhish. In natural language processing (NLP), polarity refers to the sentiment or emotion conveyed by a word or phrase in a given context. In NLP, polarity analysis is used to determine the overall sentiment of a piece of text, such as a sentence, paragraph, or entire document. This analysis can be useful in a variety of applications, such as understanding customer feedback [33].

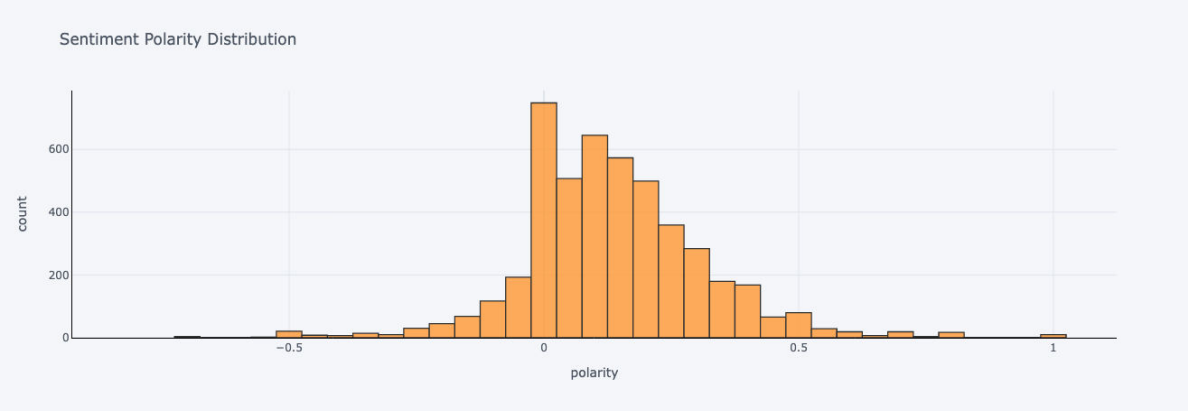


Figure 72: Sentiment Polarity Distribution in 5.1.5

Preprocessing was followed by the pipeline method. The pipeline method is a Python scikit-learn utility for orchestrating machine learning operations [34]. Pipelines function by allowing a linear series of data transforms to be linked together, resulting in a measurable modeling process [34]. The pipeline starts with a TF-IDF vectorizer, which is a measure of the originality of a word by comparing the number of times a word appears in a document with

the number of documents the word appears in. The minimum and maximum number of features in the TF-IDF vectorizer are specified, as well as the N-gram value, as seen in figure 73, which considers the relationships between frequently associated words in the word cloud.

```
# Range (inclusive) of n-gram sizes for tokenizing text.
NGRAM_RANGE = (1, 2)
#We use the top 20K features.
TOP_K = 20000
# split into word or character n-grams.
TOKEN_MODE = 'word'
# Minimum document frequency below which a token will be discarded.
MIN_DOCUMENT_FREQUENCY = 10
MAX_DOCUMENT_FREQUENCY= 4000
# Limit on the length of text sequences. Sequences longer than this will be truncated.
MAX_SEQUENCE_LENGTH = 500
stopWords =STOP_WORDS

kwargs = {
            'ngram_range': NGRAM_RANGE,  # Use 1-grams + 2-grams.
            'dtype': 'int32',
            'strip_accents': 'unicode',
            'decode_error': 'replace',
            'analyzer': TOKEN_MODE,  # Split text into word tokens.
            'min_df': MIN_DOCUMENT_FREQUENCY,
            'max_df': MAX_DOCUMENT_FREQUENCY,
            'stop_words': stopWords,
}
```

Figure 73: TF-IDF parameters in 5.1.5

The next step is Synthetic Minority Oversampling Technique (SMOTE) [35], which is an oversampling technique that solves the issue of an unbalanced dataset where there are too few examples of the minority class for a model to effectively learn the decision boundary. The minority class, which is phishing, gets oversampled by duplicating examples from that minority class in the training dataset prior to fitting a model.

Finally, we tried the training and testing again on the three AI techniques resulting in three pipelines: a SVM pipeline, a Naïve Bayes pipeline, and a Random Forest pipeline as seen in figure 74.

```
NB_pipeline = Pipeline([
                ('tfidf', TfidfVectorizer(**kwargs)),
                ('smote', SMOTE(random_state=12)),
                ('clf', MultinomialNB()),
            ])
SVM_pipeline = Pipeline([
                ('tfidf', TfidfVectorizer(**kwargs)),
                ('smote', SMOTE(random_state=12)),
                ('clf', LinearSVC(C=1)),
            ])
RF_pipeline = Pipeline([
                ('tfidf', TfidfVectorizer(**kwargs)),
                ('smote', SMOTE(random_state=12)),
                ('clf', RandomForestClassifier()),
            ])
```

Figure 74: AI algorithms' pipelines in 5.1.5

To define the performance of the three classification algorithms, the performance measures previously discussed in section 2.3 were implemented, beginning with a confusion matrix as seen in figure 75. A confusion matrix visualizes and summarizes the performance of the classification algorithm as mentioned in section 2.3.4. It showed where errors were made in the model. The rows represent the actual classes the outcomes should have been. While the columns represent the predictions we have made. Using this table, it is easy to see which predictions are wrong as visualized in figure 76, where SVM had the highest overall accurately predicted values.

```
#CONFUSION MATRIX OF NAIVE BAYES
plot_confusion_matrix(y_test, y_test_pred,title='Confusion matrix NB',
                      figsize=(4, 2), dpi=100,
                      target_names=["Phishing","Legitmate"], )

#CONFUSION MATRIX OF RANDOM FOREST
plot_confusion_matrix(y_test, y_test_pred,title='Confusion matrix RF',
                      figsize=(4, 2), dpi=100,
                      target_names=["Phishing","Legitmate"], )

#CONFUSION MATRIX OF SUPPORT VECTOR MACHINE
plot_confusion_matrix(y_test, y_test_pred,title='Confusion matrix SVM',
                      figsize=(4, 2), dpi=100,
                      target_names=["Phishing","Legitmate"], )
```

Figure 75: Confusion matrix plotting in 5.1.5

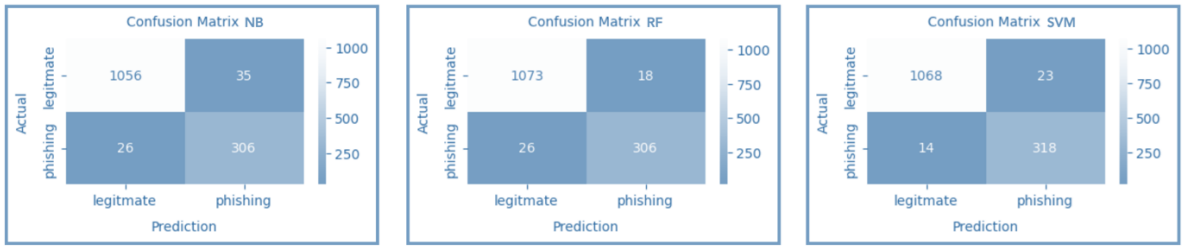


Figure 76: Confusion matrices of AI algorithms in 5.1.5

Unlike the result we got in the Orange Data Mining toolkit, the Python SVM model had the best overall performance out of the three, where the results were clearly visualized in the classification report of each AI algorithm as seen in figure 77.

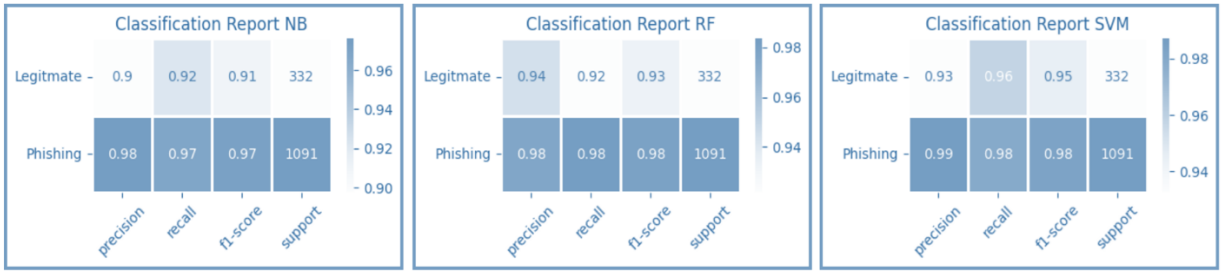


Figure 77: Classification report in 5.1.5

The accuracy of the three classification algorithms was calculated as well as seen in Figure 78. It is important for CyberPhish to calculate the accuracy, to ensure reliable results and the model that has the higher accurate performance is SVM as shown in table 8.

```
NB_pipeline.fit(X_train, y_train)
y_test_pred = NB_pipeline.predict(X_test)
accuracy = accuracy_score(y_test,y_test_pred)*100

SVM_pipeline.fit(X_train, y_train)
y_test_pred = SVM_pipeline.predict(X_test)
accuracy = accuracy_score(y_test,y_test_pred)*100

RF_pipeline.fit(X_train, y_train)
y_test_pred = RF_pipeline.predict(X_test)
accuracy = accuracy_score(y_test,y_test_pred)*100
```

Figure 78: Python Accuracy Calculations in 5.1.5

Table 8: Accuracy Results in 5.1.5

Algorithm	Accuracy
Naïve Bayes	95.71
Random Forest	96.91
SVM	97.40

The Python SVM model was then tested on the API to ensure the integration went well. Then the model got deployed to the server to be tested on a real-life email inbox using the CyberPhish application. The inbox had 100 emails, 95 being legitimate and 5 being phishy emails. The optimal results were when 1 out of 100 emails was misclassified. And the testing continued in more than one Gmail account with good results each test.

5.1.6 AI Troubleshooting

CyberPhish faced another issue which was because of the dataset that was originally created. The solution to this issue was recommended to the CyberPhish team by Dr. Mohammad Almukaynizi during a bootcamp called “AI in Cybersecurity”, where he mentioned that to create a closer to life representation of the user’s inbox, the data should be unbalanced. The problem with the unbalanced data then was solved using SMOTE.

5.2 User Acceptance Testing

In this section, we will present the user acceptance testing tables for each of the twenty test participants as shown below from table 9 to table 28, which illustrate their interaction with the CyberPhish application. We will also provide an overview of the demographics of the test participants, followed by an analysis of their testing experiences based on the questionnaire data collected.

As for the testing process for CyberPhish’s application features, starting with ‘login using Gmail’, this aims to calculate CyberPhish’s response time since the user clicks on the login button until the user is directed to the home page, excluding the time while the user fills out their account information and verification that’s done by the Google API.

Then the user explores the application and calculates the time needed to navigate to each page. Further, an email should be sent to the user's inbox to calculate the syncing of a

newly received email. Also, an email should be permanently deleted from the user's inbox, and the time it takes to reflect that change on CyberPhish's application.

After that, the user interacts with the chatbot, and the time of this task will depend on the tester's willingness to complete the quiz. Finally, the user has to log out of the application.

Table 9: Tester 1 Results in 5.2

Tester 1			
Task	Duration	pass	comment
1. Login using Gmail address.	30s	Yes	
2. Explore home (inbox) screen.	8s	Yes	
3. Explore specific legitimate email and its components.	12s	Yes	
4. Read specific phishing email and its components.	17s	Yes	
5. Syncing new received email.	4s	Yes	
6. Syncing after deleting an email.	3s	Yes	
7. Navigate to the awareness screen.	2s	Yes	
8. Choose a certain article to view/read.	3s	Yes	
9. View the article in the web	2s	Yes	
10. **Navigate to the report screen.**	1s	Yes	
11. View and understand data displayed in "This year" report.	10s	Yes	
12. View and understand data displayed in "This month" report.	6s	Yes	
13. View and understand data displayed in "This week" report.	7s	Yes	
14. Navigate to the chatbot.	1s	Yes	
15. Go back to the main menu in the chatbot.	3	Yes	
16. Interact with the quiz provided in the chatbot.	3 min	Yes	
17. Interact with the FAQ provided in the chatbot.	1 min	Yes	
18. Sign out	10s	No	Tester didn't find the sign-out button

Table 10: Tester 2 Results in 5.2

Tester 2			
Task	Duration	pass	comment
1. Login using Gmail address.	26s	Yes	
2. Explore home (inbox) screen.	19s	Yes	
3. Explore specific legitimate email and its components.	12s	Yes	

4. Read specific phishing email and its components.	23s	Yes	
5. Syncing new received email.	8s	Yes	
6. Syncing after deleting an email.	3s	Yes	
7. Navigate to the awareness screen.	2s	Yes	
8. Choose a certain article to view/read.	4s	Yes	
9. View the article in the web	5s	Yes	
10. Navigate to the report screen.	3s	Yes	
11. View and understand data displayed in "This year" report.	6s	Yes	
12. View and understand data displayed in "This month" report.	5s	Yes	
13. View and understand data displayed in "This week" report.	5s	Yes	
14. Navigate to the chatbot.	1s	Yes	
15. Go back to the main menu in the chatbot.	3s	Yes	
16. Interact with the quiz provided in the chatbot.	2min	Yes	
17. Interact with the FAQ provided in the chatbot.	8s	Yes	
18. Sign out	5s	Yes	

Table 11: Tester 3 Results in 5.2

Tester 3			
Task	Duration	pass	comment
1. Login using Gmail address.	25s	Yes	
2. Explore home (inbox) screen.	8s	Yes	
3. Explore specific legitimate email and its components.	10s	Yes	
4. Read specific phishing email and its components.	14s	Yes	
5. Syncing new received email.	10s	Yes	
6. Syncing after deleting an email.	4s	Yes	
7. Navigate to the awareness screen.	10s	Yes	
8. Choose a certain article to view/read.	7s	Yes	
9. View the article in the web	4s	Yes	
10. **Navigate to the report screen.**	4s	Yes	
11. View and understand data displayed in "This year" report.	8s	Yes	
12. View and understand data displayed in "This month" report.	3s	Yes	
13. View and understand data displayed in "This week" report.	5s	Yes	
14. Navigate to the chatbot.	3s	Yes	
15. Go back to the main menu in the chatbot.	4s	Yes	
16. Interact with the quiz provided in the chatbot.	20s	Yes	
17. Interact with the FAQ provided in the chatbot.	10s	Yes	
18. Sign out	5s	Yes	

Table 12: Tester 4 Results in 5.2

Tester 4			
Task	Duration	pass	comment
1. Login using Gmail address.	15s	Yes	
2. Explore home (inbox) screen.	10s	Yes	
3. Explore specific legitimate email and its components.	12s	Yes	

4. Read specific phishing email and its components.	30s	Yes	
5. Syncing new received email.	9s	Yes	
6. Syncing after deleting an email.	3s	Yes	
7. Navigate to the awareness screen.	10s	Yes	
8. Choose a certain article to view/read.	30s	Yes	
9. View the article in the web	8s	Yes	
10. **Navigate to the report screen.**	3s	Yes	
11. View and understand data displayed in "This year" report.	14s	Yes	
12. View and understand data displayed in "This month" report.	20s	Yes	
13. View and understand data displayed in "This week" report.	10s	Yes	
14. Navigate to the chatbot.	4s	Yes	
15. Go back to the main menu in the chatbot.	1s	Yes	
16. Interact with the quiz provided in the chatbot.	34s	Yes	
17. Interact with the FAQ provided in the chatbot.	10s	Yes	
18. Sign out	4s	Yes	

Table 13: Tester 5 Results in 5.2

Tester 5			
Task	Duration	pass	comment
1. Login using Gmail address.	20s	Yes	
2. Explore home (inbox) screen.	16s	Yes	
3. Explore specific legitimate email and its components.	16s	Yes	
4. Read specific phishing email and its components.	9s	Yes	
5. Syncing new received email.	11s	Yes	
6. Syncing after deleting an email.	5s	Yes	
7. Navigate to the awareness screen.	9s	No	Tester was not able to recognize awareness screen icon.
8. Choose a certain article to view/read.	15s	Yes	
9. View the article in the web	6s	Yes	
10. **Navigate to the report screen.**	7s	Yes	
11. View and understand data displayed in "This year" report.	24s	Yes	
12. View and understand data displayed in "This month" report.	15s	Yes	
13. View and understand data displayed in "This week" report.	12s	Yes	
14. Navigate to the chatbot.	6s	Yes	
15. Go back to the main menu in the chatbot.	5s	Yes	
16. Interact with the quiz provided in the chatbot.	12s	Yes	
17. Interact with the FAQ provided in the chatbot.	15s	Yes	
18. Sign out	5s	Yes	

Table 14: Tester 6 Results in 5.2

Tester 6			
Task	Duration	pass	comment
1. Login using Gmail address.	10s	Yes	
2. Explore home (inbox) screen.	20s	Yes	
3. Explore specific legitimate email and its components.	10s	Yes	
4. Read specific phishing email and its components.	12s	Yes	
5. Syncing new received email.	8s	Yes	
6. Syncing after deleting an email.	3s	Yes	
7. Navigate to the awareness screen.	4s	Yes	
8. Choose a certain article to view/read.	20s	Yes	
9. View the article in the web	3s	Yes	
10. **Navigate to the report screen.**	2s	Yes	
11. View and understand data displayed in "This year" report.	20s	Yes	
12. View and understand data displayed in "This month" report.	10s	Yes	
13. View and understand data displayed in "This week" report.	7s	Yes	

14. Navigate to the chatbot.	8s	Yes	
15. Go back to the main menu in the chatbot.	2s	Yes	
16. Interact with the quiz provided in the chatbot.	15s	Yes	
17. Interact with the FAQ provided in the chatbot.	7s	Yes	
18. Sign out	10s	Yes	

Table 15: Tester 7 Results in 5.2

Tester 7			
Task	Duration	pass	comment
1. Login using Gmail address.	8s	Yes	
2. Explore home (inbox) screen.	15s	Yes	
3. Explore specific legitimate email and its components.	7s	Yes	
4. Read specific phishing email and its components.	8s	Yes	
5. Syncing new received email.	13s	Yes	
6. Syncing after deleting an email.	3s	Yes	
7. Navigate to the awareness screen.	3s	Yes	
8. Choose a certain article to view/read.	10s	Yes	
9. View the article in the web	3s	Yes	
10. **Navigate to the report screen.**	3s	Yes	
11. View and understand data displayed in "This year" report.	16s	Yes	
12. View and understand data displayed in "This month" report.	8s	Yes	
13. View and understand data displayed in "This week" report.	10s	Yes	
14. Navigate to the chatbot.	3s	Yes	
15. Go back to the main menu in the chatbot.	2s	Yes	
16. Interact with the quiz provided in the chatbot.	20s	Yes	
17. Interact with the FAQ provided in the chatbot.	4s	Yes	
18. Sign out	5s	Yes	

Table 16: Tester 8 Results in 5.2

Tester 8			
Task	Duration	pass	comment
1. Login using Gmail address.	25s	Yes	
2. Explore home (inbox) screen.	12s	Yes	
3. Explore specific legitimate email and its components.	10s	Yes	
4. Read specific phishing email and its components.	15s	Yes	
5. Syncing new received email.	9s	Yes	
6. Syncing after deleting an email.	3s	Yes	
7. Navigate to the awareness screen.	2s	Yes	
8. Choose a certain article to view/read.	8s	Yes	
9. View the article in the web	3s	Yes	
10. **Navigate to the report screen.**	3s	Yes	
11. View and understand data displayed in "This year" report.	2s	Yes	
12. View and understand data displayed in "This month" report.	20s	Yes	
13. View and understand data displayed in "This week" report.	4s	Yes	
14. Navigate to the chatbot.	13s	Yes	
15. Go back to the main menu in the chatbot.	2s	Yes	
16. Interact with the quiz provided in the chatbot.	1:43s	Yes	
17. Interact with the FAQ provided in the chatbot.	9s	Yes	
18. Sign out	3s	Yes	

Table 17: Tester 9 Results in 5.2

Tester 9			
Task	Duration	pass	comment
1. Login using Gmail address.	29s	Yes	

2.	**Explore home (inbox) screen.**	12s	Yes	
3.	**Explore specific legitimate email and its components.**	10s	Yes	
4.	**Read specific phishing email and its components.**	14s	Yes	
5.	**Syncing new received email.**	7s	Yes	
6.	**Syncing after deleting an email.**	4s	Yes	
7.	**Navigate to the awareness screen.**	5s	Yes	
8.	**Choose a certain article to view/read.**	15s	Yes	
9.	**View the article in the web**	5s	Yes	
10.	**Navigate to the report screen.**	3s	Yes	
11.	**View and understand data displayed in "This year" report.**	10s	Yes	
12.	**View and understand data displayed in "This month" report.**	7s	Yes	
13.	**View and understand data displayed in "This week" report.**	8s	Yes	
14.	**Navigate to the chatbot.**	4s	Yes	
15.	**Go back to the main menu in the chatbot.**	3s	Yes	
16.	**Interact with the quiz provided in the chatbot.**	17s	Yes	
17.	**Interact with the FAQ provided in the chatbot.**	5s	Yes	
18.	**Sign out**	3s	Yes	

Table 18: Tester 10 Results in 5.2

Tester 10			
Task	Duration	pass	comment
1. Login using Gmail address.	27s	Yes	
2. Explore home (inbox) screen.	40s	Yes	
3. Explore specific legitimate email and its components.	15s	Yes	
4. Read specific phishing email and its components.	18s	Yes	
5. Syncing new received email.	12s	Yes	
6. Syncing after deleting an email.	4s	Yes	
7. Navigate to the awareness screen.	2s	Yes	
8. Choose a certain article to view/read.	10s	Yes	
9. View the article in the web	4s	Yes	
10. **Navigate to the report screen.**	1s	Yes	
11. View and understand data displayed in "This year" report.	15s	Yes	
12. View and understand data displayed in "This month" report.	3s	Yes	
13. View and understand data displayed in "This week" report.	5s	Yes	
14. Navigate to the chatbot.	4s	Yes	
15. Go back to the main menu in the chatbot.	1	Yes	
16. Interact with the quiz provided in the chatbot.	33s	Yes	
17. Interact with the FAQ provided in the chatbot.	12s	Yes	
18. Sign out	4s	Yes	

Table 19: Tester 11 Results in 5.2

Tester 11			
Task	Duration	pass	comment
1. Login using Gmail address.	17s	Yes	
2. Explore home (inbox) screen.	7s	Yes	
3. Explore specific legitimate email and its components.	9s	Yes	
4. Read specific phishing email and its components.	12s	Yes	
5. Syncing new received email.	11s	Yes	
6. Syncing after deleting an email.	5s	Yes	
7. Navigate to the awareness screen.	2s	Yes	
8. Choose a certain article to view/read.	5s	Yes	
9. View the article in the web	*2s*	Yes	
10. **Navigate to the report screen.**	2s	Yes	
11. View and understand data displayed in "This year" report.	2s	Yes	
12. View and understand data displayed in "This month" report.	1s	Yes	
13. View and understand data displayed in "This week" report.	1s	Yes	

14. Navigate to the chatbot.	4s	Yes	
15. Go back to the main menu in the chatbot.	1s	Yes	
16. Interact with the quiz provided in the chatbot.	34s	Yes	
17. Interact with the FAQ provided in the chatbot.	7s	Yes	
18. Sign out	5s	Yes	

Table 20: Tester 12 Results in 5.2

Tester 12			
Task	Duration	pass	comment
1. Login using Gmail address.	20s	Yes	
2. Explore home (inbox) screen.	10s	Yes	
3. Explore specific legitimate email and its components.	6s	Yes	
4. Read specific phishing email and its components.	8s	Yes	
5. Syncing new received email.	13s	Yes	
6. Syncing after deleting an email.	6s	Yes	
7. Navigate to the awareness screen.	4s	Yes	
8. Choose a certain article to view/read.	13s	Yes	
9. View the article in the web	5s	Yes	
10. **Navigate to the report screen.**	4s	Yes	
11. View and understand data displayed in "This year" report.	24s	Yes	
12. View and understand data displayed in "This month" report.	12s	Yes	
13. View and understand data displayed in "This week" report.	10s	Yes	
14. Navigate to the chatbot.	4s	Yes	
15. Go back to the main menu in the chatbot.	5s	Yes	
16. Interact with the quiz provided in the chatbot.	31s	Yes	
17. Interact with the FAQ provided in the chatbot.	15s	Yes	
18. Sign out	5s	Yes	

Table 21: Tester 13 Results in 5.2

Tester 13			
Task	Duration	pass	comment
1. Login using Gmail address.	20s	Yes	
2. Explore home (inbox) screen.	13s	Yes	
3. Explore specific legitimate email and its components.	15s	Yes	
4. Read specific phishing email and its components.	30s	Yes	
5. Syncing new received email.	13s	Yes	
6. Syncing after deleting an email.	5s	Yes	
7. Navigate to the awareness screen.	4s	Yes	
8. Choose a certain article to view/read.	10s	Yes	
9. View the article in the web	3s	Yes	
10. **Navigate to the report screen.**	3s	Yes	
11. View and understand data displayed in "This year" report.	10s	Yes	
12. View and understand data displayed in "This month" report.	13s	Yes	
13. View and understand data displayed in "This week" report.	8s	Yes	
14. Navigate to the chatbot.	4s	Yes	
15. Go back to the main menu in the chatbot.	2s	Yes	
16. Interact with the quiz provided in the chatbot.	14s	Yes	
17. Interact with the FAQ provided in the chatbot.	10s	Yes	
18. Sign out	5s	Yes	

Table 22: Tester 14 Results in 5.2

Tester 14			
Task	Duration	pass	comment
1. Login using Gmail address.	30s	Yes	

2. Explore home (inbox) screen.	10s	Yes	
3. Explore specific legitimate email and its components.	9s	Yes	
4. Read specific phishing email and its components.	12s	Yes	
5. Syncing new received email.	10s	Yes	
6. Syncing after deleting an email.	6s	Yes	
7. Navigate to the awareness screen.	7s	Yes	
8. Choose a certain article to view/read.	12s	Yes	
9. View the article in the web	3s	Yes	
10. **Navigate to the report screen.**	4s	Yes	
11. View and understand data displayed in "This year" report.	10s	Yes	
12. View and understand data displayed in "This month" report.	9s	Yes	
13. View and understand data displayed in "This week" report.	7s	Yes	
14. Navigate to the chatbot.	10s	Yes	
15. Go back to the main menu in the chatbot.	3s	Yes	
16. Interact with the quiz provided in the chatbot.	12s	Yes	
17. Interact with the FAQ provided in the chatbot.	10s	Yes	
18. Sign out	8s	No	Tester was not able to find the logout button.

Table 23: Tester 15 Results in 5.2

Tester 15			
Task	Duration	pass	comment
1. Login using Gmail address.	17s	Yes	
2. Explore home (inbox) screen.	6s	Yes	
3. Explore specific legitimate email and its components.	5s	Yes	
4. Read specific phishing email and its components.	8s	Yes	
5. Syncing new received email.	19s	Yes	
6. Syncing after deleting an email.	6s	Yes	
7. Navigate to the awareness screen.	2s	Yes	
8. Choose a certain article to view/read.	10s	Yes	
9. View the article in the web	2s	Yes	
10. **Navigate to the report screen.**	1s	Yes	
11. View and understand data displayed in "This year" report.	1s	Yes	
12. View and understand data displayed in "This month" report.	1s	Yes	
13. View and understand data displayed in "This week" report.	1s	Yes	
14. Navigate to the chatbot.	1s	Yes	
15. Go back to the main menu in the chatbot.	1s	Yes	
16. Interact with the quiz provided in the chatbot.	44s	Yes	
17. Interact with the FAQ provided in the chatbot.	1s	Yes	
18. Sign out	2s	Yes	

Table 24: Tester 16 Results in 5.2

Tester 16			
Task	Duration	pass	comment
1. Login using Gmail address.	14s	Yes	
2. Explore home (inbox) screen.	26s	Yes	
3. Explore specific legitimate email and its components.	8s	Yes	
4. Read specific phishing email and its components.	12s	Yes	
5. Syncing new received email.	15s	Yes	

6. Syncing after deleting an email.	5s	Yes	
7. Navigate to the awareness screen.	6s	Yes	
8. Choose a certain article to view/read.	10s	Yes	
9. View the article in the web	4s	Yes	
10. **Navigate to the report screen.**	1s	Yes	
11. View and understand data displayed in "This year" report.	1s	Yes	
12. View and understand data displayed in "This month" report.	1s	Yes	
13. View and understand data displayed in "This week" report.	1s	Yes	
14. Navigate to the chatbot.	1s	Yes	
15. Go back to the main menu in the chatbot.	1s	Yes	
16. Interact with the quiz provided in the chatbot.	14s	Yes	
17. Interact with the FAQ provided in the chatbot.	1s	Yes	
18. Sign out	1s	Yes	

Table 25: Tester 17 Results in 5.2

Tester 17			
Task	Duration	pass	comment
1. Login using Gmail address.	8s	Yes	
2. Explore home (inbox) screen.	17s	Yes	
3. Explore specific legitimate email and its components.	7s	Yes	
4. Read specific phishing email and its components.	8s	Yes	
5. Syncing new received email.	13s	Yes	
6. Syncing after deleting an email.	4s	Yes	
7. Navigate to the awareness screen.	2s	Yes	
8. Choose a certain article to view/read.	8s	Yes	
9. View the article in the web	1s	Yes	
10. **Navigate to the report screen.**	1s	Yes	
11. View and understand data displayed in "This year" report.	1s	Yes	
12. View and understand data displayed in "This month" report.	1s	Yes	
13. View and understand data displayed in "This week" report.	1s	Yes	
14. Navigate to the chatbot.	2s	Yes	
15. Go back to the main menu in the chatbot.	1s	Yes	
16. Interact with the quiz provided in the chatbot.	30s	Yes	
17. Interact with the FAQ provided in the chatbot.	1s	Yes	
18. Sign out	1s	Yes	

Table 26: Tester 18 Results in 5.2

Tester 18			
Task	Duration	pass	comment
1. Login using Gmail address.	9s	Yes	
2. Explore home (inbox) screen.	21s	Yes	
3. Explore specific legitimate email and its components.	3s	Yes	
4. Read specific phishing email and its components.	10s	Yes	
5. Syncing new received email.	18s	Yes	
6. Syncing after deleting an email.	5s	Yes	
7. Navigate to the awareness screen.	2s	Yes	
8. Choose a certain article to view/read.	11s	Yes	
9. View the article in the web	4s	Yes	
10. **Navigate to the report screen.**	1s	Yes	
11. View and understand data displayed in "This year" report.	1s	Yes	
12. View and understand data displayed in "This month" report.	1s	Yes	
13. View and understand data displayed in "This week" report.	1s	Yes	
14. Navigate to the chatbot.	1s	Yes	
15. Go back to the main menu in the chatbot.	1s	Yes	

16. Interact with the quiz provided in the chatbot.	27s	Yes	
17. Interact with the FAQ provided in the chatbot.	1s	Yes	
18. Sign out	3s	Yes	

Table 27: Tester 19 Results in 5.2

Tester 19			
Task	Duration	pass	comment
1. Login using Gmail address.	6s	Yes	
2. Explore home (inbox) screen.	23s	Yes	
3. Explore specific legitimate email and its components.	12s	Yes	
4. Read specific phishing email and its components.	26s	Yes	
5. Syncing new received email.	14s	Yes	
6. Syncing after deleting an email.	5s	Yes	
7. Navigate to the awareness screen.	2s	Yes	
8. Choose a certain article to view/read.	14s	Yes	
9. View the article in the web	2s	Yes	
10. **Navigate to the report screen.**	1s	Yes	
11. View and understand data displayed in "This year" report.	1s	Yes	
12. View and understand data displayed in "This month" report.	1s	Yes	
13. View and understand data displayed in "This week" report.	1s	Yes	
14. Navigate to the chatbot.	2s	Yes	
15. Go back to the main menu in the chatbot.	2s	Yes	
16. Interact with the quiz provided in the chatbot.	44s	Yes	
17. Interact with the FAQ provided in the chatbot.	2s	Yes	
18. Sign out	7s	No	Tester was not able to recognize logout icon.

Table 28: Tester 20 Results in 5.2

Tester 20			
Task	Duration	pass	comment
1. Login using Gmail address.	10s	Yes	
2. Explore home (inbox) screen.	36s	Yes	
3. Explore specific legitimate email and its components.	11s	Yes	No indication of scroll ability
4. Read specific phishing email and its components.	15s	Yes	
5. Syncing new received email.	11s	Yes	
6. Syncing after deleting an email.	6s	Yes	
7. Navigate to the awareness screen.	2s	Yes	
8. Choose a certain article to view/read.	17s	Yes	
9. View the article in the web	2s	Yes	
10. **Navigate to the report screen.**	15s	No	
11. View and understand data displayed in "This year" report.	1s	Yes	
12. View and understand data displayed in "This month" report.	1s	Yes	
13. View and understand data displayed in "This week" report.	1s	Yes	
14. Navigate to the chatbot.	2s	Yes	
15. Go back to the main menu in the chatbot.	1s	Yes	
16. Interact with the quiz provided in the chatbot.	47s	Yes	
17. Interact with the FAQ provided in the chatbot.	1s	Yes	
18. Sign out	38s	No	Tester was not able to recognize logout icon.

5.2.1 Demographics of Participants

In this section, we present an overview of the demographic characteristics of the participants who took part in the user acceptance testing of CyberPhish. This information is crucial for understanding the user feedback and experiences gathered during the testing phase.

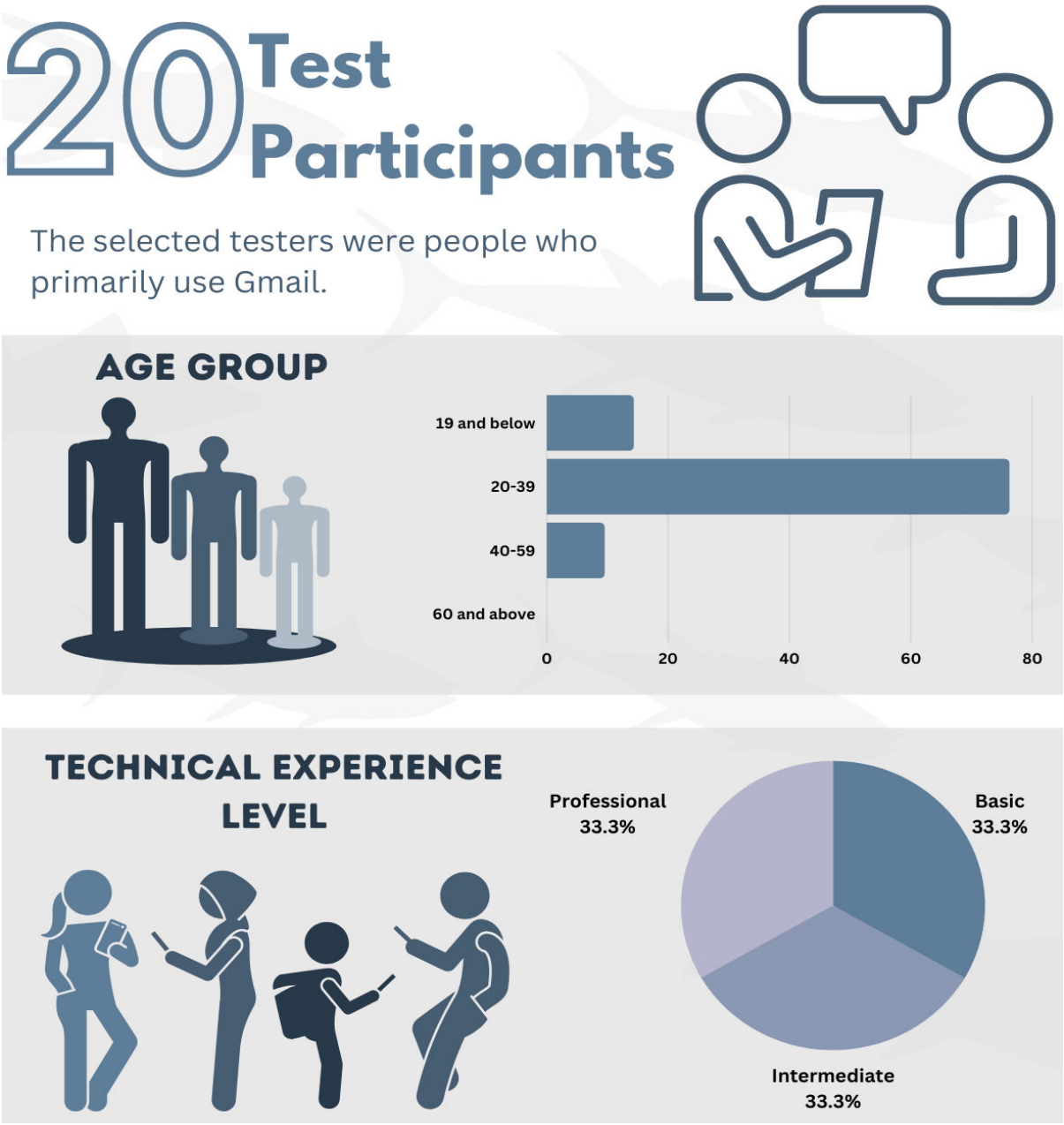


Figure 79: UAT Participant Demographics in 5.2.1

5.2.2 Questionnaire/Interview Results

We conducted a user acceptance test on twenty Gmail users using our application CyberPhish. To gain feedback on their experience, we prepared a questionnaire (see Appendix B: UAT Questionnaire) with fifteen questions. The results are as follows:

Figure 80 illustrates that the test participants were from different age ranges, with the majority of participants being in the 20-39 age group with a percentage of 76.2%.

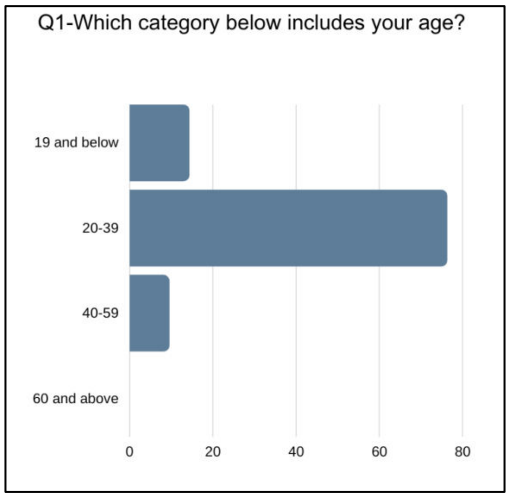


Figure 80: Survey 1st Question in 5.2.2

Furthermore, the percentage in Figure 81 reflects the participants' degree of technical skill and experience, and the results were balanced, with all levels having a 33.3% percentage. This demonstrates that the participants had variable degrees of technical skill, allowing us to observe how users with differing levels of technical expertise might respond to the app and offer us with various points of view and opinions on the application.

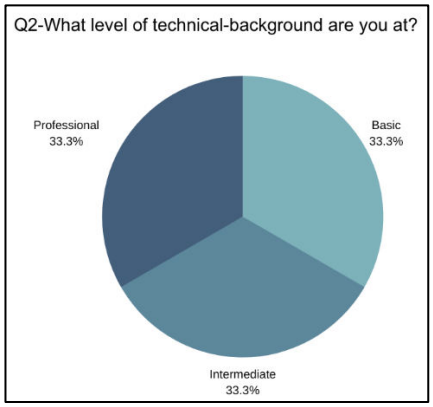


Figure 81: Survey 2nd Question in 5.2.2

When asked if CyberPhish does its main job of detecting phishing emails, the majority of the participants strongly agreed (81%), and some participants agreed (19%) as shown in Figure 82.

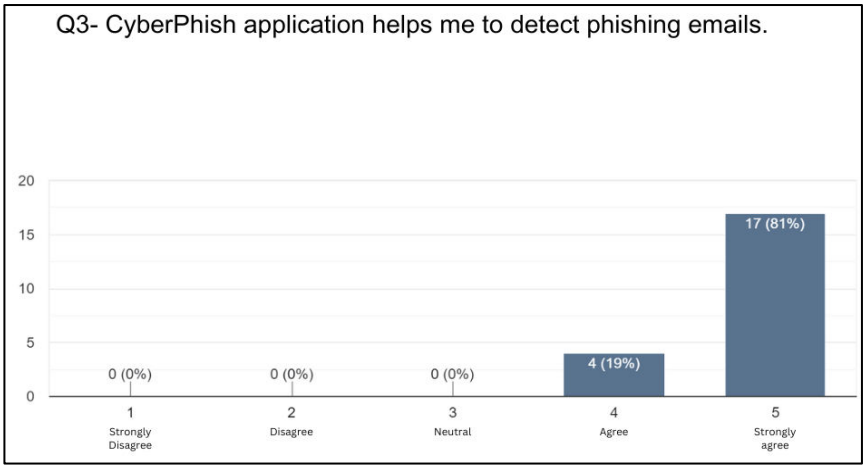


Figure 82: Survey 3rd Question in 5.2.2

From Figure 83, we can see that 76.2% of the participants strongly agree that it is straightforward to receive and view incoming emails using the CyberPhish application, while 14.3% agree and 9.5% are neutral.

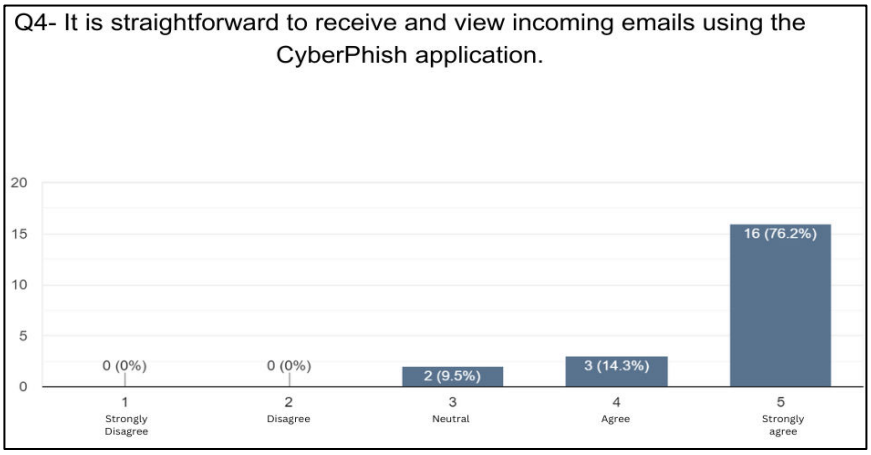


Figure 83: Survey 4th Question in 5.2.2

As shown in Figure 84, the majority of participants gave positive feedback regarding the risk score percentages, with 81% strongly agreeing and 19% agreeing that it was clear and easy to understand. This shows that the application's scoring system is well understood by all users of varying technical expertise.

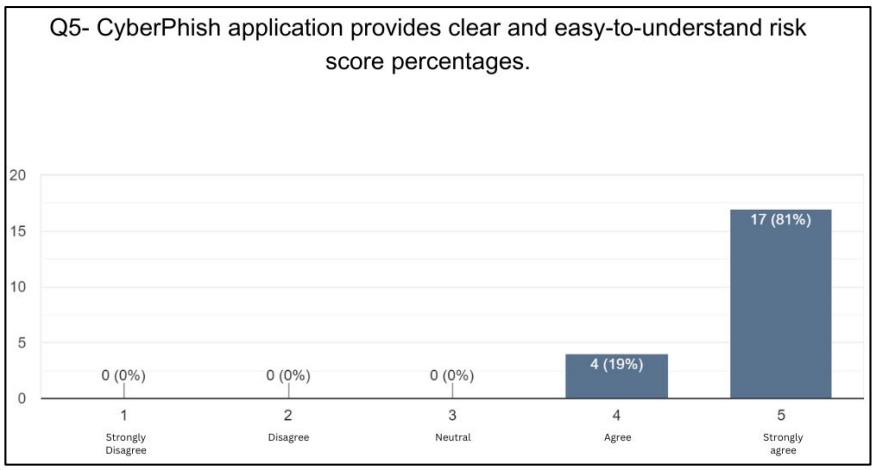


Figure 84: Survey 5th Question in 5.2.2

The risk score percentages are accompanied by an explanation and reasoning for emails flagged as phishing, and when asked about it, 85.7% of the participants strongly agree that they found it to be understandable, and 14.3% agree, as shown in Figure 85.

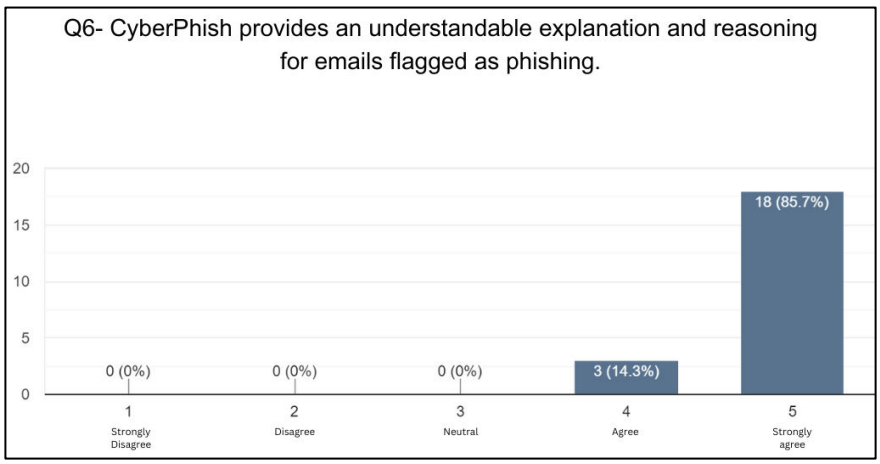


Figure 85: Survey 6th Question in 5.2.2

Moving on to the analytical reports page, the participants were asked whether they thought that the data presented was useful and informative. As illustrated in Figure 86, 90.5% of the participants think it's very useful, while 9.5% think it's useful.

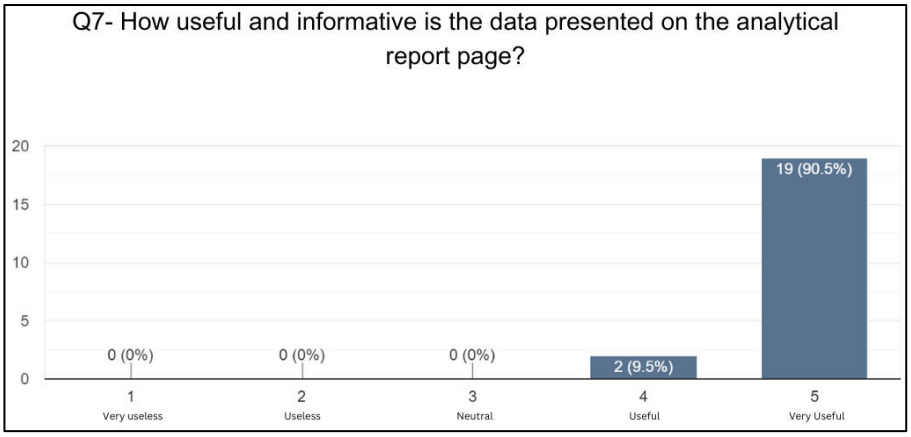


Figure 86: Survey 7th Question in 5.2.2

Following that question, the participants were also asked how easy it was to customize the analytical report to display the specific time frame they wanted. By observing Figure 87, it is clear that most users find it easy to customize the analytical report, with 81% thinking it's very easy and 19% thinking it's easy.

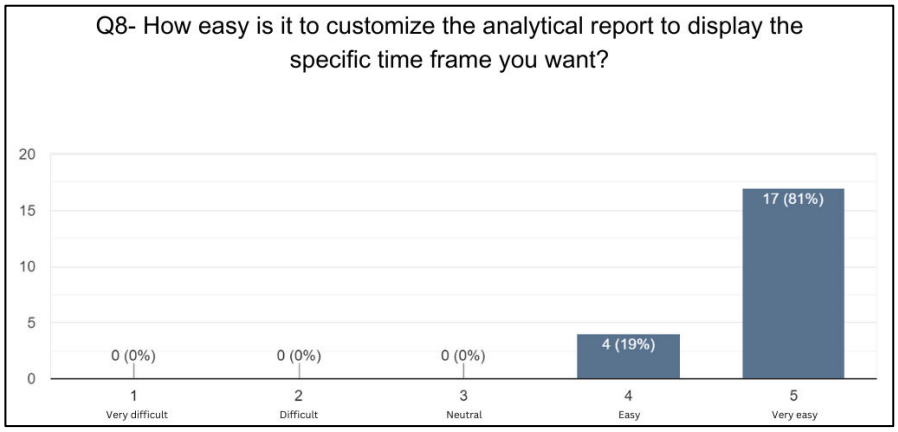


Figure 87:Survey 8th Question in 5.2.2

Next is the awareness content page, where the participants were asked if they thought the content provided helped raise their knowledge about phishing emails. 85.7% strongly agree, while 14.3% agree, as shown in Figure 88.

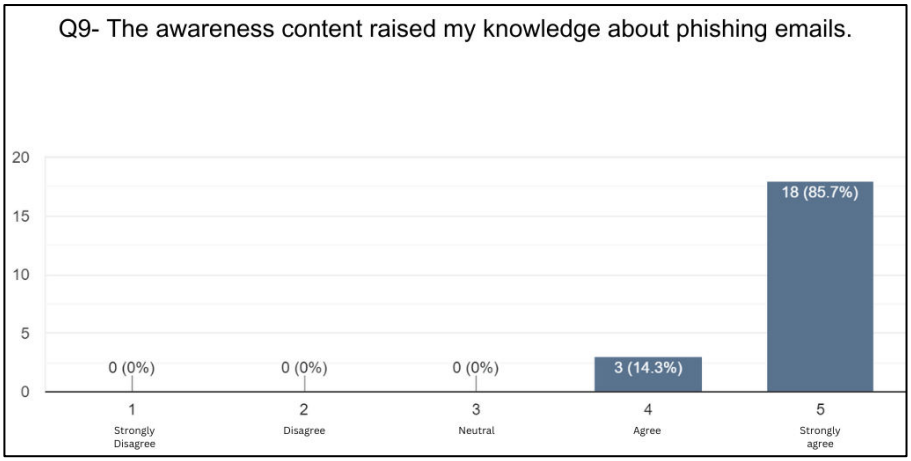


Figure 88: Survey 9th Question in 5.2.2

Next, the participants were asked if they found the chatbot game to be fun and insightful, to which 85.7% strongly agreed and 14.3% agreed, as seen in Figure 89.

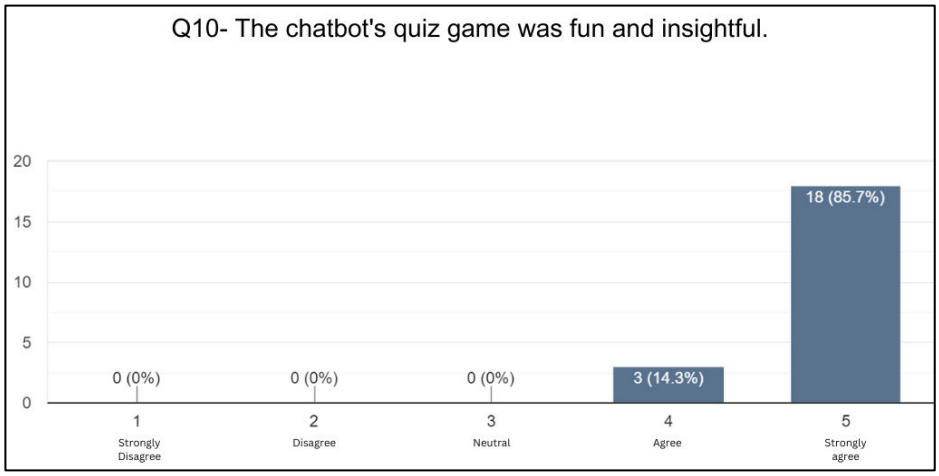


Figure 89: Survey 10th Question in 5.2.2

The participants were then asked if they thought that the interface was friendly and easy to navigate through; 71.1% strongly agreed, 23.8% agreed, and 4.8% were neutral, as shown in Figure 90.

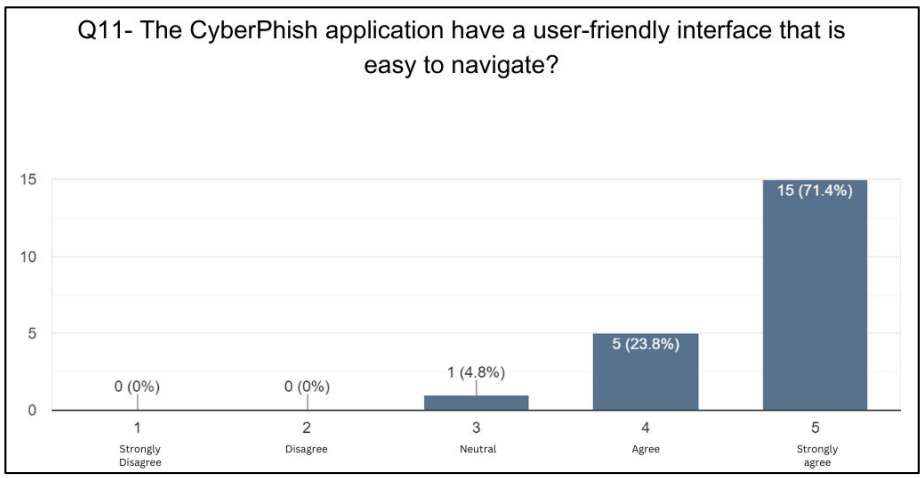


Figure 90: Survey 11th Question in 5.2.2

When asked about the application's response speed, the participants gave varying results, with 42.9% voting for very fast, 42.9% voting for fast, and 14.3% voting for moderate, as illustrated in Figure 91.

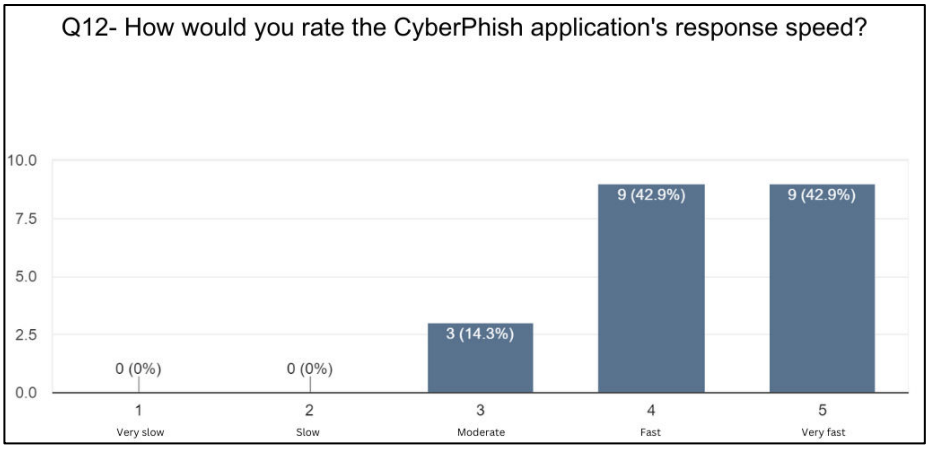


Figure 91: Survey 12th Question in 5.2.2

When questioned about the helpfulness of the CyberPhish app's features, 95.2% of the participants strongly agreed, while 4.8% agreed as shown in Figure 92.

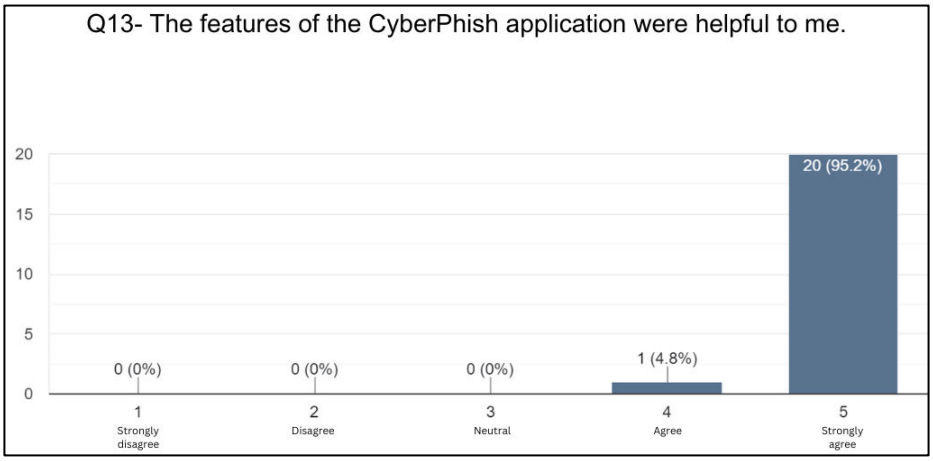


Figure 92: Survey 13th Question in 5.2.2

The participants were asked to rate their overall experience with the CyberPhish application, to which 85.7% answered excellent and 14.3% answered good (see Figure 93).

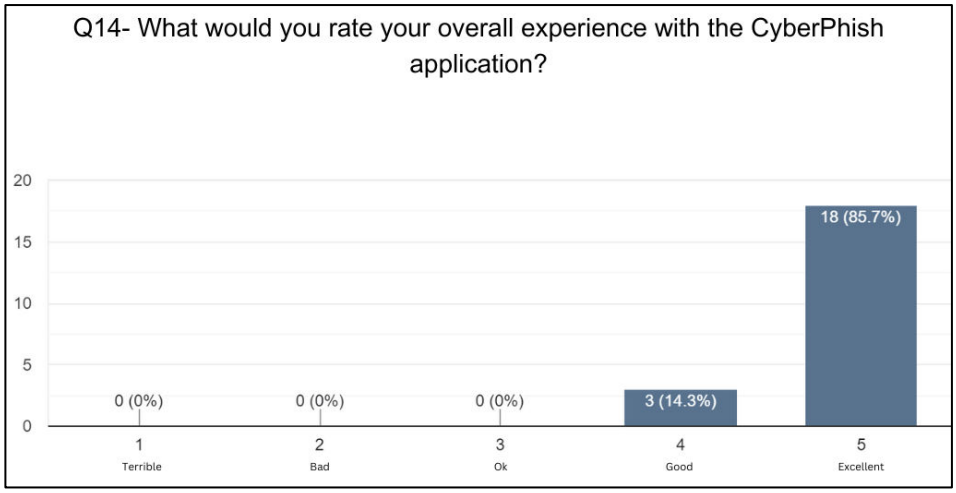


Figure 93: Survey 14th Question in 5.2.2

Finally, the participants were asked to provide recommendations for improving CyberPhish As seen in Figure 94. We received a number of responses, including suggestions to improve processing speed, build a Chrome extension, make the sign-out button larger and clearer, ensure that the labels at the bottom navigation bar are always visible, support the Arabic language, and make it more obvious that users can scroll down in emails.

Figure 94: Survey 15th Question in 5.2.2

5.3 Quality Attributes (NFR testing)

In this section, we explained how CyberPhish's non-functional testing has been done and its results as shown in Table 29.

Table 29: Quality Attributes (NFR testing) in 5.3

User story	Quality Attribute	Measure	Results
As a CyberPhish user, I want the application to be available 99% of the time, so that I will not miss a chance to detect a phishing email.	Availability: Whether CyberPhish is operational, functional and usable for fulfilling the user's requirements at any desired time.	Test the functionality of CyberPhish on different users, at different times of a day, and at the same time. CyberPhish was always available for all users whether they request it at the same time or at	•We gathered a total of 20 testers. •First case, we requested each one of them to try the app in a different time/day. •Second case, we did the test on each pair at the same time. •Third case, we did

		different times a day.	the testing on 6 of them one after another during a short period of time. •In all three cases, all testers were able to access CyberPhish services.
As a CyberPhish user, I want the feedback to be displayed with no delay having a stable internet connection, so that I don't waste my time.	Performance: How fast does the system returns results?	Compute the time it takes to display the classified email. The system needs less than 11 seconds in average to classify an email.	•When a user logs in, we start a timer to calculate the amount of time to display a classified email. •When user receives the email, we stop the timer and document the result. •20 users have completed the test. •The average time was 11 seconds, maximum 19 seconds, and minimum 4 seconds.
As a CyberPhish user, I want the application	Usability: How easy is the	Compute the time it takes the CyberPhish	•We start a timer for each task the user

to be simple to use, so that I do not make mistakes and waste time and energy learning how to use the application.	application's interface to use and understand by the user?	user to use all the application functions. The user needs less than 7 minutes to use the application functions.	must do and document the results. •20 users have completed the test. •the average total time spent to interact with CyberPhish was 7 minutes, maximum 14 minutes, and minimum 5 minutes.
As a CyberPhish user, I want my sessions to be inactivated and destroyed after I log out, so that no one can hijack my session.	Security: How well are the system and its data protected against attacks?	The CyberPhish user session must be inactivated and destroyed which means no one can use the user's session in CyberPhish application.	• All the 20 users' session expired after logging out from the application. • The maximum response time was in 13 seconds, the average response time was in 6 seconds and the minimum response time was in 1 second.

5.4 Discussion

In this subsection, we will provide an interpretation of the results presented in the previous sections. In general, the results of the system evaluation phase were deemed satisfactory. First of all, by looking at the user acceptance tables offered in User Acceptance Testing we can note that we completed the majority of our non-functional needs and their acceptance criteria. Furthermore, the majority of the functions were completed by the users without mistakes and in a reasonable amount of time. Furthermore, we provided a questionnaire to the testers in order for them to offer feedback on their experience with CyberPhish, and we reviewed the results under Questionnaire/Interview Results. According to the responses collected, all testers were pleased with the application's interface and the functions that we tested.

In addition, some testers gave us important input on how we might enhance our application. the feedback we received regarding the homepage pertained to testers' preference for replacing the logout icon with something clearer accompanied by the words "log out" to help that users locate the button. The testers raised concerns that an unskilled user could be unable to distinguish between all of the pages on the bottom navigation bar, and that the page labels should be visible and apparent at all times. Additional feedback provided by the testers includes suggestions to incorporate a vertical scroll bar for displaying lengthy email content and enhancements to the application's processing speed, as well as adding support for the Arabic language.

6 Conclusions and Future Work

In this section we will talk about the conclusion, which is a bridge to help our readers make the transition back to their daily lives. This section will help visualize why the development effort is worthwhile.

This document represents our journey with CyberPhish, starting from the thought, and growing this thought over all stages we walked through, starting from the introduction chapter which clarifies the idea and gives a general introduction of CyberPhish. The introduction chapter is followed by the background chapter which has an important role in preparing the reader to understand the CyberPhish details, by providing a brief explanation of knowledge aspects that CyberPhish falls in, such as cyberattacks, what is phishing, and AI in phishing detection. To deliver an application that fills the gap in the applications market, and to specify CyberPhish features, we reviewed and discussed academic papers and mobile applications in the same field of CyberPhish and were represented in the literature review chapter. Once we had an envisage of CyberPhish features, we started system analysis and design chapter which transforms CyberPhish features into a form used to facilitate the implementation of the CyberPhish application and support the understanding of some CyberPhish components and how they interact with each other. After we analyzed our system, we started developing CyberPhish using Flutter framework and testing it to ensure it's free from bugs. Our idea came to life to add value to society and just like a seed doesn't instantly become a mature plant, ideas take time to develop.

6.1 Global and Local Impact.

6.1.1 Local Impact

As the Kingdom of Saudi Arabia’s movement towards digital transformation rises, so does the dangers of unsafe email communications. CyberPhish will assist Saudis in better understanding which email communications are legitimate, and which are fraudulent. Also, CyberPhish will help users in having better understanding and defense against the attacks that threaten the security of their daily communications.

6.1.2 Global Impact

In our busy lives, email communications became more frequent, but many people tend to pay less attention when it comes to the small details in their emails. these details are often

a tell-tale sign of a phishing attack. CyberPhish solves this problem, where we pay attention for the minor details, so the users do not have to, by considering the development of artificial intelligence technologies and utilizing them to develop the solution. As the world discusses the cyberspace safety strategies, they must take into consideration the communication safety of the individuals in this cyberspace.

6.2 Problems and Challenges

6.2.1 Implementation

CyberPhish's implementation had a four-month long issue with the display of the extracted email body. CyberPhish began parsing the body of the email and displaying it as text only, but the problem with parsing the email body was that it didn't display the body as it is designed and intended, and with no attachments, or images which did not do the email any justice look wise. Then, putting the text in a web page, meaning that the email body opens a web page in the user's browser, and that did not work as well and messed the application. Then CyberPhish tried to showcase the email as text as well as extract the images to display them with the email body. This method although worked as an idea, but it did not look right since it was disordered, with different sizing, and other design issues. Then this method was tried again but an order was implemented where the text comes first, then the link if any, then the images if any, but it still did not look as how the emails look by email service providers. Lastly, after trials and errors, CyberPhish tried to not parse the HTML email body but instead store the HTML body as it is and send it to display the HTML email body to the user using the web view widget and also display any image type so it looks like the emails other email services provide. Moreover, the plaintext and links in email bodies are parsed to be used when analyzing the emails.

6.2.2 Training and Testing

Initially, the aim was to employ Weka during the model training step. The attempts to use Weka resulted in failure since the dataset was in CSV format and had to be converted to ARFF, but the conversion processes did not work. At this point, the search for an alternative software began, and while we were looking, we began working on an SVM model in Python code. The dataset was inserted into the Python code SVM model and then split 70/30. After that, the now-split dataset was sent into the word vectorizer, followed by training the SVM model, where the results and the matrix have been shown. The Python code adventure came

to an end when we discovered an alternative software called Orange Data Mining. While using Orange Data Mining we tried Leave-One-Out cross-validation but abandoned it because it was ineffective with the dataset due to the fact that each row in the dataset had no relationships to any other rows, so we only used K-Fold cross-validation.

6.3 Limitations of the System

When we started working on the Gmail API, we created a Gmail API account so we could define the project, take the credentials, and then give a predefined list of email accounts that were allowed to access.

After that, we tried using the code and it immediately granted us access to all the emails and their fields, like the header, subject, message, protocol, etc. Then when we tried to publish it so all of us could try it on our own laptops, Gmail asked for verification. This verification is sent after we make a formal request. In this request we were asked about three things: justification; abiding by their data privacy policy, and a security assessment.

In the justification, we were supposed to explain what our project was and why we needed this type of access because email access is restricted. As for the data privacy policy, we were to abide by their policy and to explain that through a YouTube video. The security assessment would be done by Google where they would look over the code and our requirements.

The problem with this is that it takes weeks for this process to be completed and it costs 40,000 SAR. The only limitation with this was that we cannot access just anybody's email inbox, and the predefined list of testing email accounts that was mentioned in the beginning can only have 100 email accounts.

Moreover, CyberPhish used an external system called APIVoid to analyze the sender's email reputation and URL links reputation. This system only allows three requests per second with which causes the application's response time some delay. We looked for alternatives, but other systems had less requests per second which left us with APIVoid as the better choice.

6.4 Main Contribution of the Project

We currently live in a digital age where the market has been impacted by mobile apps. A significant shift in how individuals "enjoy" amenities has been brought about by the most

recent trend in mobile app development. CyberPhish improves a person's life by ensuring secure email communication.

6.5 Future Work

Looking towards the future, the CyberPhish team is committed to expanding the project's reach and impact. One key area of focus is making the project more widely accessible, including making it available in Arabic to cater to a broader audience. Additionally, the team plans to develop a version of the project specifically designed for organizational emails to enhance its effectiveness in corporate settings.

Another area of focus is making the project more accessible to IOS users. The team plans to optimize the project for IOS devices, ensuring that it is user-friendly and provides seamless protection against phishing attacks.

Furthermore, the CyberPhish team is committed to making the project more widely available to the public by deploying it on popular app stores such as the App Store and Google Play. This will allow users to easily download and install CyberPhish, ensuring that they are protected from phishing attempts on their mobile devices.

By implementing these strategies, the CyberPhish team is confident that the project will become even more effective in combatting phishing attacks. The CyberPhish team remains committed to ongoing development and improvement, seeking to provide a comprehensive and user-friendly solution to the growing threat of phishing attacks.

7 Achievements

Participating in competitions is an essential aspect of the development process for any project, and CyberPhish has taken full advantage of these opportunities to showcase its capabilities and potential. CyberPhish has garnered numerous participations in competitions around the Middle East. In March 2023, CyberPhish won the third place in Cyberthon Biban23, given by Thakaa Center of the General Authority for Small and Medium Enterprises. This achievement has been a tremendous honor and a source of pride for the CyberPhish team, validating the hard work and dedication put into the project and serving as a significant motivator for the CyberPhish team, fueling the passion and drive to continue improving and innovating.

In addition to winning an award, CyberPhish is also participating in various competitions, further showcasing its capabilities and potential. CyberPhish is currently competing in several competitions, including the 10th Undergraduate Research Competition (URC), where it will showcase its innovative approach to combating phishing attacks. It has been nominated to present CyberPhish in the final round at Abu Dhabi University, which is a testament to the project's potential and impact.

Furthermore, CyberPhish will participate in the 13th annual Scientific Forum, held by King Saud University, and is qualified for the next stage of the forum. These participations provide a platform for CyberPhish to demonstrate its effectiveness and receive valuable feedback from experts in the field. The feedback and recognition received from these events have and will continue to inspire the CyberPhish team to continue developing and improving on CyberPhish, ultimately leading to its continued success.

8 Acknowledgements

We would like to express our gratitude to everyone who has supported us throughout our graduation project. I am especially thankful to Dr. Alia Alabdulkarim and Dr. Nora Alhammad, our supervisors, for their guidance, academic encouragement, and friendly critique. Their knowledge, expertise, attitude, and care have been invaluable in helping us to develop our skills and to complete our graduation project “CyberPhish” as planned and on time.

We are also grateful to our families and friends for their unwavering support and encouragement. Without their help, we would not have been able to complete this project. Finally, I would like to thank faculty members at King Saud University, for providing us with the resources and support we needed to complete this project. Thank you all for helping us reach this milestone.

9 References

[1]. ITgovernance, "What is Phishing? Attack Techniques & Prevention Tips," ITgovernance, [Online]. Available: https://www.itgovernance.co.uk/phishing
[2]. S. Widup, M. Spitler, D. Hylender and G. Bassett, "2018 Verizon Data Breach Investigations Report.".
[3]. ThreatLabz, "2022 ThreatLabz Phishing Report. San Jose, California: Zscaler."
[4]. DHL Replaces Microsoft as Most Imitated Brand in Phishing Attempts in Q4 2021," [Online]. Available: https://blog.checkpoint.com/2022/01/17/dhl-replaces-microsoft-as-most-imitated-brand-in-phishing-attempts-in-q4-2021/.
[5]. javapoint, "SVM," [Online]. Available: https://www.javatpoint.com/
[6]. cisco, "Cyber security threat trends: phishing, crypto top the list," cisco, 2021.
[7]. Proofpoint, "What Is Phishing?," [Online]. Available: https://www.proofpoint.com/us/threat-reference/phishing.
[8]. Kaspersky, "What is Social Engineering?," [Online]. Available: https://www.kaspersky.com/resource-center/definitions/what-is-social-engineering.
[9]. R. Raj, "E-mail spam and non-spam filtering using Machine Learning," [Online]. Available: https://www.enjoyalgorithms.com/blog/email-spam-and-non-spam-filtering-using-machine-learning
[10]. www.analyticsvidhya.com," [Online]. Available: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/
[11]. www.orangedatamining.com," [Online]. Available: https://orangedatamining.com/widget-catalog/model/svm/
[12]. www.v7labs.com," [Online]. Available: https://www.v7labs.com/blog/f1-score-guide#:~:text=F1%20score%20is%20a%20machine%20learning%20evaluation%20metric%20that%20measures,prediction%20across%20the%20entire%20dataset
[13]. www.analyticsvidhya.com," [Online]. Available: https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning/
[14]. www. towardsdatascience.com," [Online]. Available: https://towardsdatascience.com/8-metrics-to-measure-classification-performance-984d9d7fd7aa
[15]. www. machinelearningmastery.com," [Online]. Available: https://machinelearningmastery.com/k-fold-cross-validation/
[16]. www. machinelearningmastery.com," [Online]. Available: https://machinelearningmastery.com/loocv-for-evaluating-machine-learning-algorithms/
[17]. Google. (n.d.). Gmail API. Retrieved from Gmail for Developers : https://developers.google.com/gmail/api/guides
[18]. www. orangedatamining.com," [Online]. Available: https://orangedatamining.com
[19]. "www. APIVoid.com," [Online]. Available: https://www.apivoid.com
[20]. "www. kommunicate.io," [Online]. Available: https://www.kommunicate.io
[21]. "https://cloud.google.com/dialogflow [Online]. Available: https://cloud.google.com/dialogflow
[22]. Anti-Phishing Software. (n.d.). Retrieved from Avanan: https://www.avanan.com/anti-phishing-software

[23]. Intelligent Email Security . (n.d.). Retrieved from Cofense: https://cofense.com/
[24]. Mimecast. Retrieved from Mimecast: https://www.mimecast.com/
[25]. PhishTector - Gmail Phishing Detector. (n.d.). Retrieved from Chrome webstore: https://shortest.link/4EKQ
[26]. emailveritas.comop, "Phishing Detector," 31 January 2022. [Online]. Available: https://workspace.google.com/marketplace/app/phishing_detector/729413434097?hl=en.
[27]. e.Veritas, "Email Veritas Phishing Detector," [Online]. Available: https://www.emailveritas.com/phishing-detector#Advanced-Threat-Analysis.
[28]. www. GitHub.com," [Online]. Available: https://en.wikipedia.org/wiki/GitHub
[29]. Britannica, T. Editors of Encyclopaedia (2021, May 13). client-server architecture. Encyclopedia Britannica. https://www.britannica.com/technology/client-server-architecture
[30]. Sarangam, A. (2020, December 30). What Is Client Server Architecture? An Overview. Retrieved from UNext Jigsaw: https://shortest.link/4EI9
[31]. www. analyticsvidhya.com," [Online]. Available: https://www.analyticsvidhya.com/blog/2021/08/quick-hacks-to-save-machine-learning-model-using-pickle-and-joblib/#:~:text=In%20Python%2C%20the%20%E2%80%9Cpickle%E2%80%9D,retrain%20the%20model%20from%20scratch.
[32]. Nguyen, Quang Hung, et al. "Influence of data splitting on performance of machine learning models in prediction of shear strength of soil." *Mathematical Problems in Engineering* 2021 (2021).
[33]. Hutto, C. J., & Gilbert, E. (2014). VADER: A parsimonious rule-based model for sentiment analysis of social media text. Eighth international AAAI conference on weblogs and social media.
[34]. “www.askpython.com," [Online]. Available: https://www.askpython.com/python/examples/pipelining-in-python#:~:text=The%20pipeline%20is%20a%20Python,in%20a%20measurable%20modeling%20process
[35]. “www.machinelearningmastery.com," [Online]. Available: https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/
[36]. OWASP, "PHISHING IN DEPTH," [Online]. Available: https://owasp.org/www-chapter-ghana/assets/slides/OWASP_Presentation_FINAL.pdf.
[37]. https://youtu.be/82ETlrSTHBE

10 Appendix

10.1 Appendix A: Questionnaire

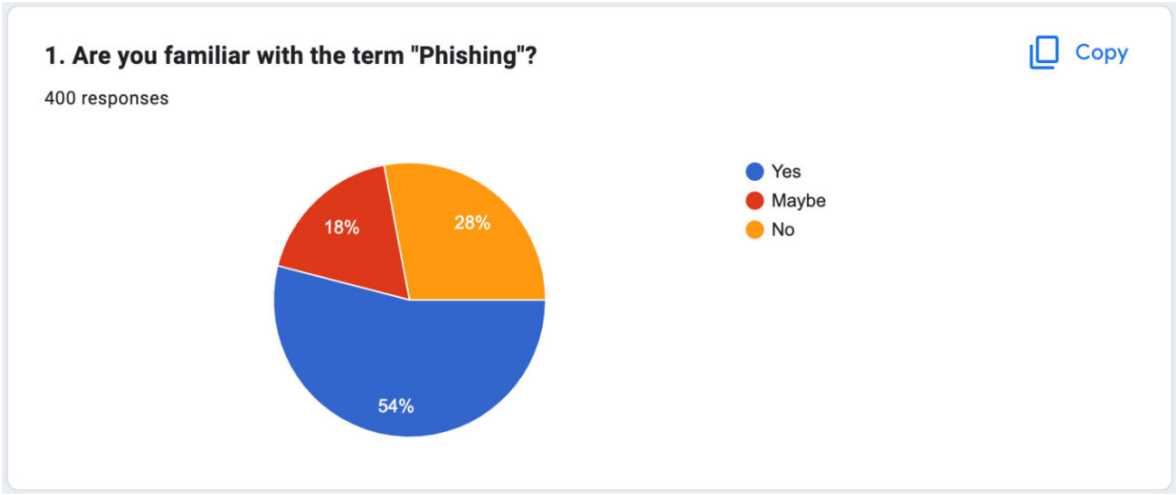


Figure 95: Question 1 in 10.1

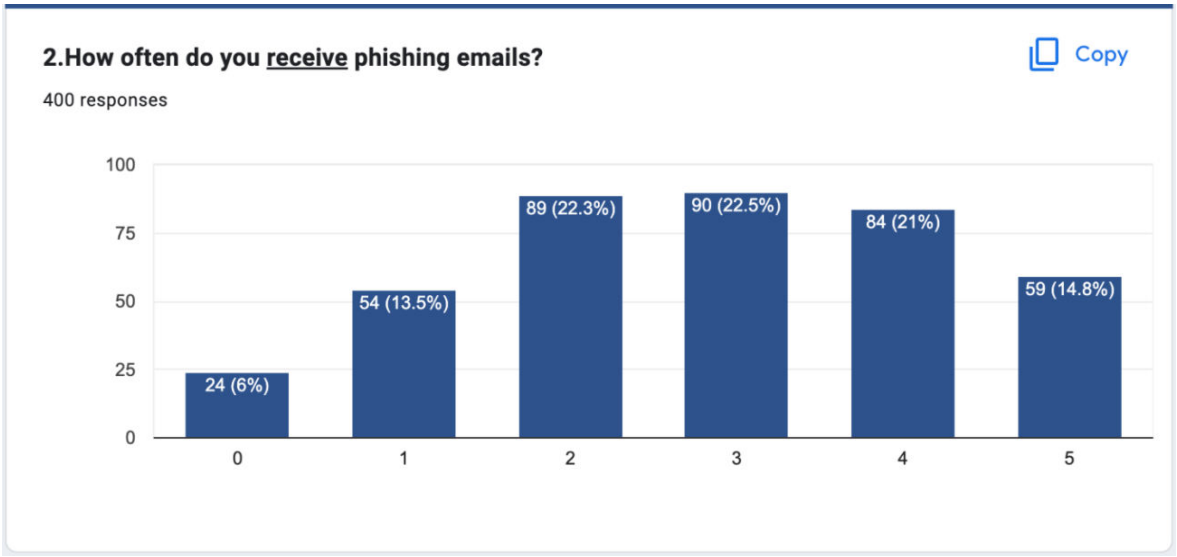


Figure 96 : Question 2 in 10.1

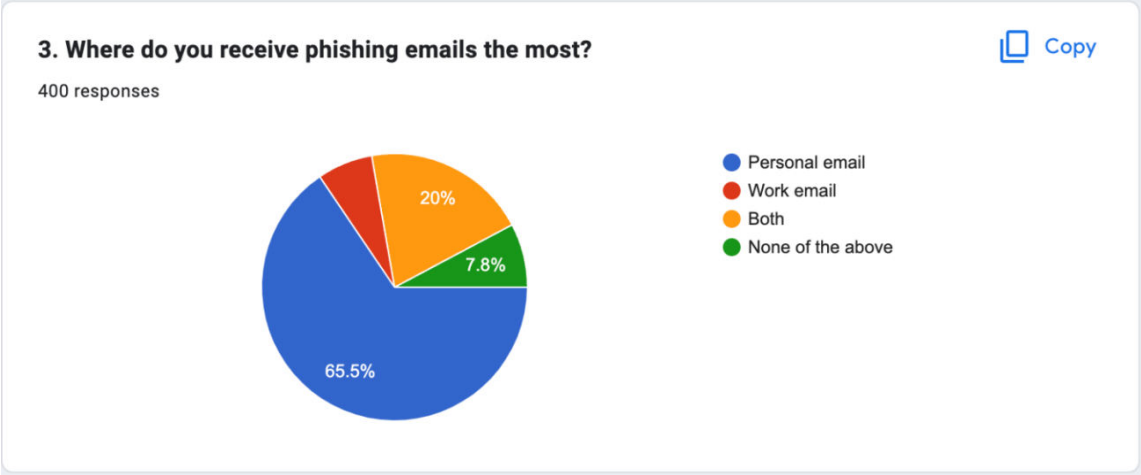


Figure 97: Question 3 in 10.1

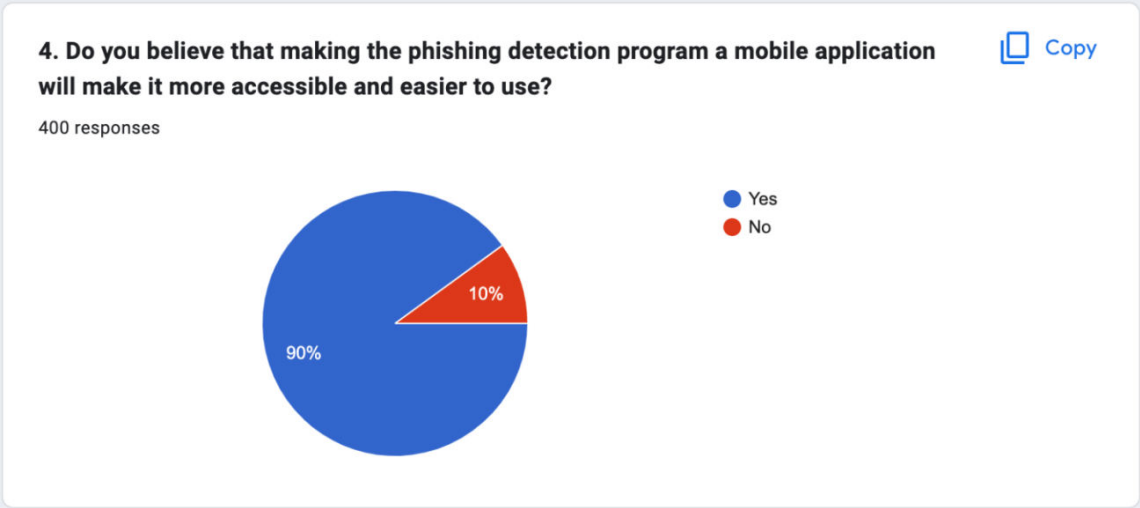


Figure 98: Question 4 in 10.1

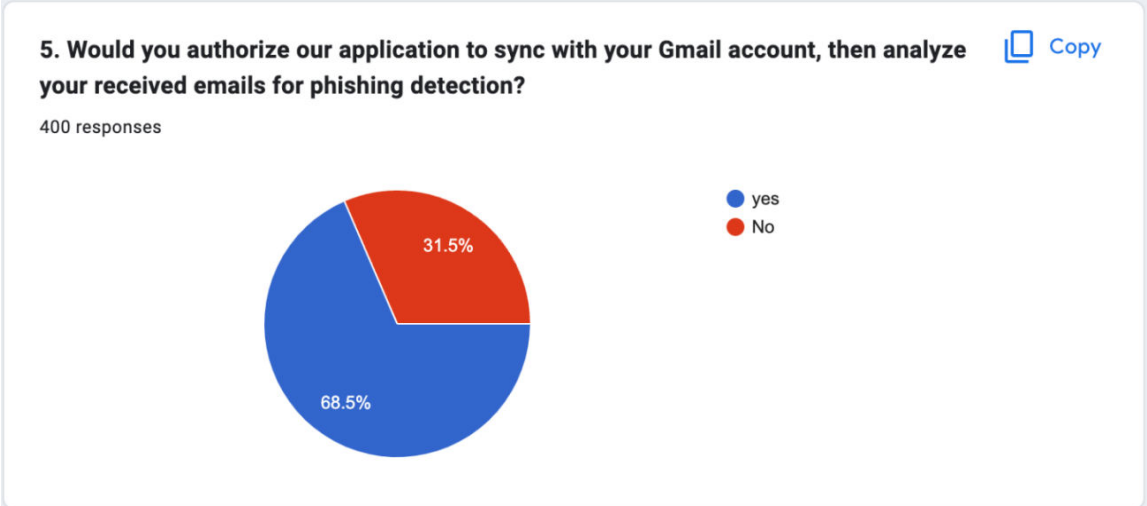


Figure 99 : Question 5 in 10.1

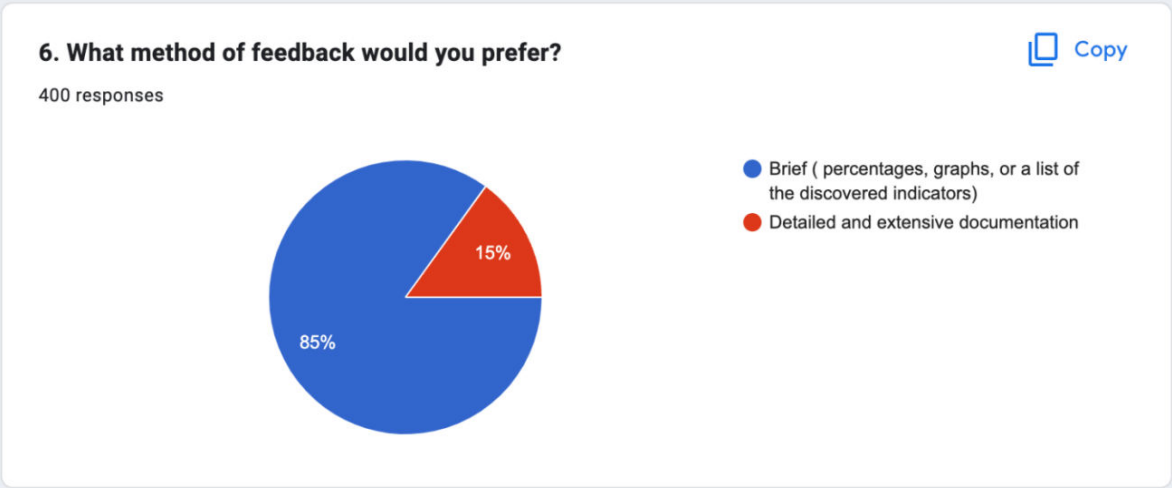


Figure 100: Question 6 in 10.1

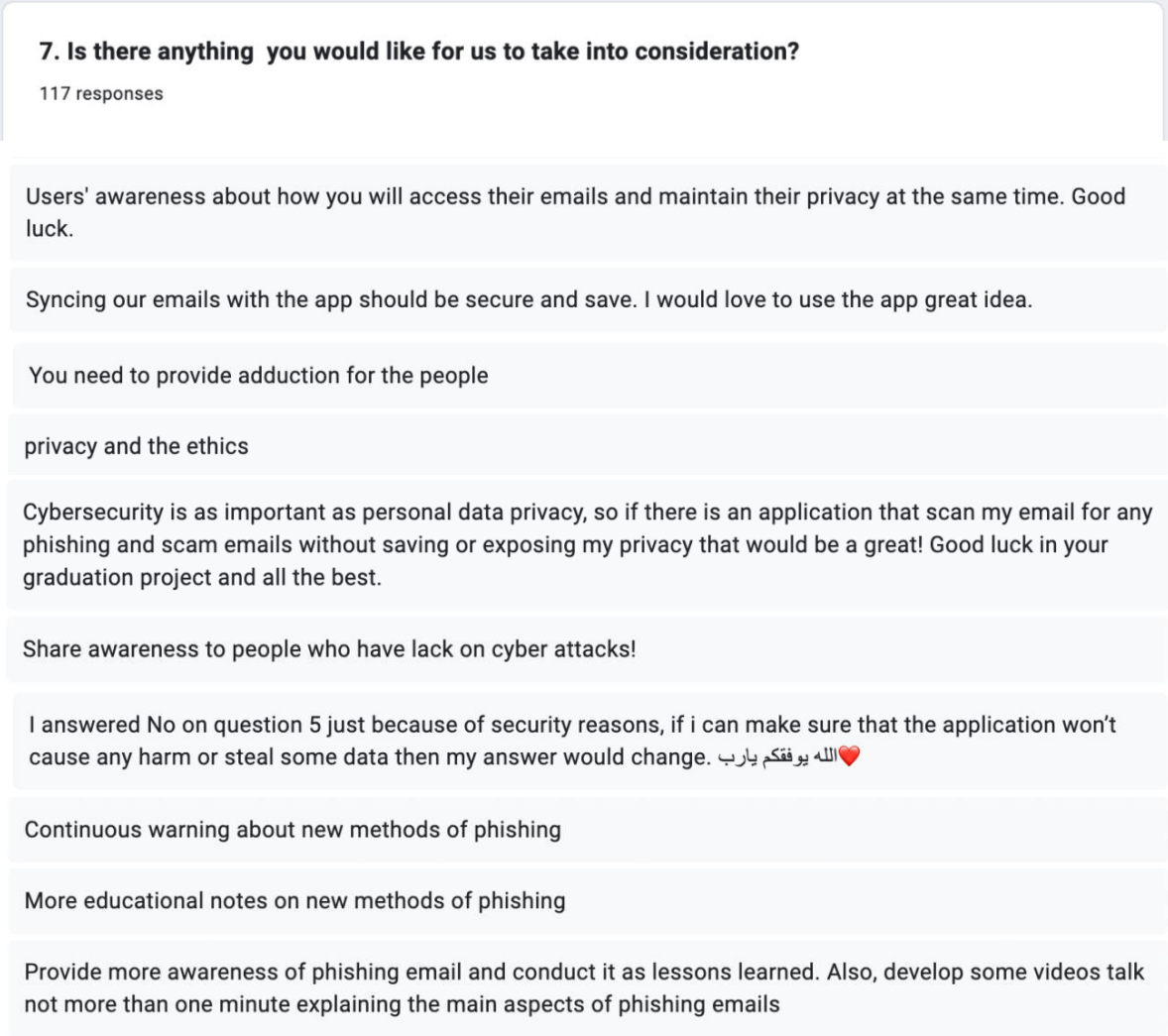


Figure 101: Question 7 in 10.1

10.2 Appendix B: UAT Questionnaire

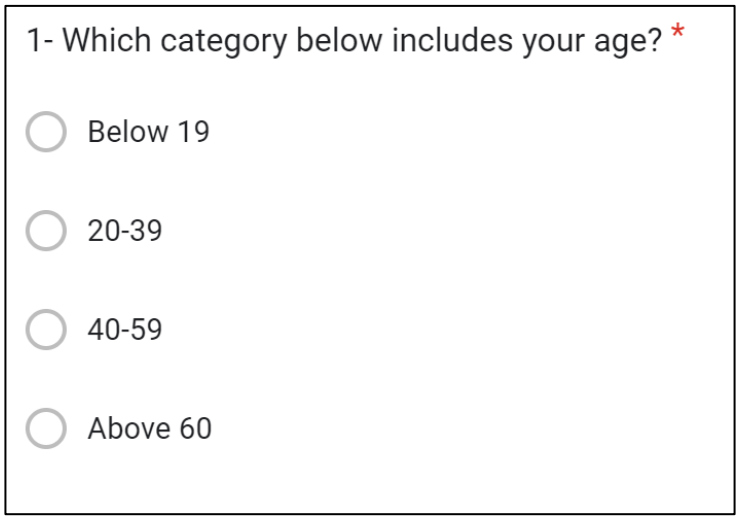


Figure 102: Question 1 in 10.2

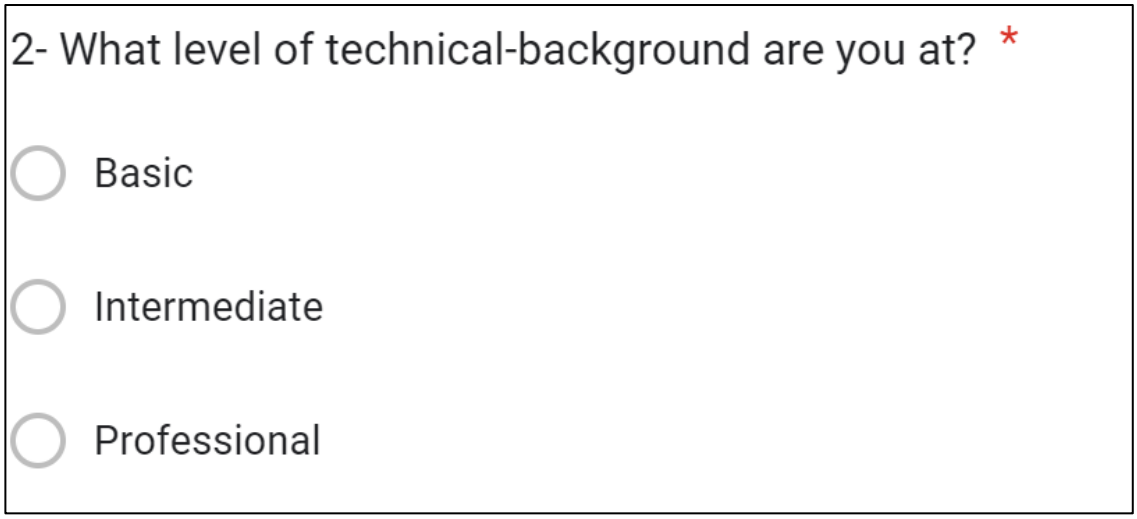


Figure 103: Question 2 in 10.2

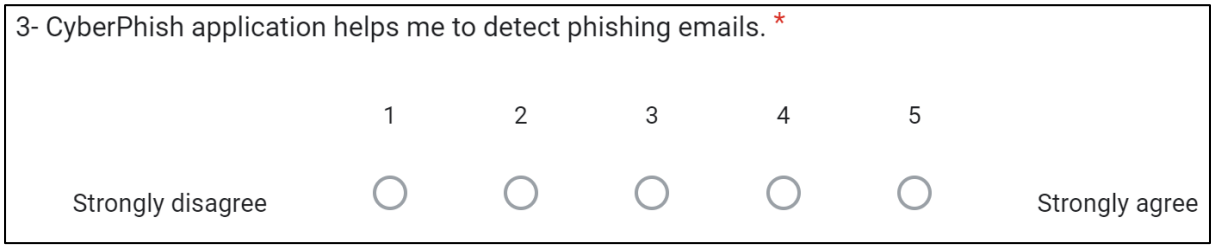


Figure 104: Question 3 in 10.2

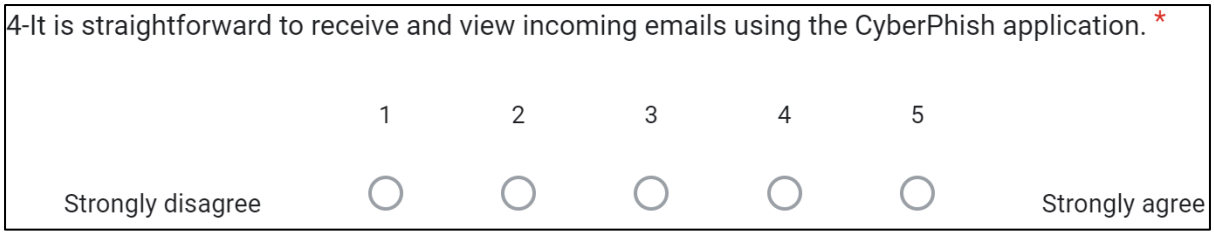


Figure 105 : Question 4 in 10.2

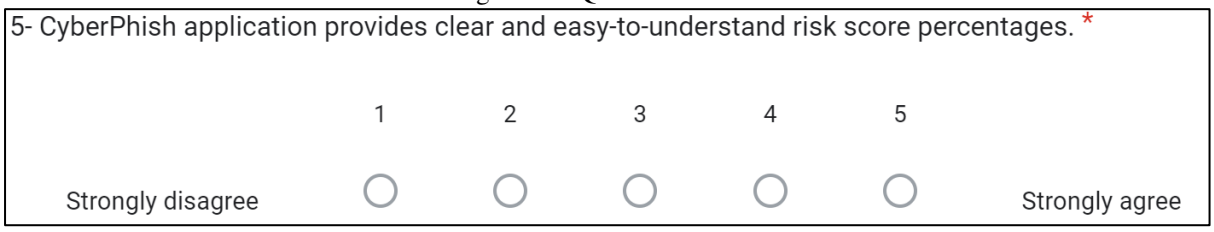


Figure 106:Question 5 in 10.2

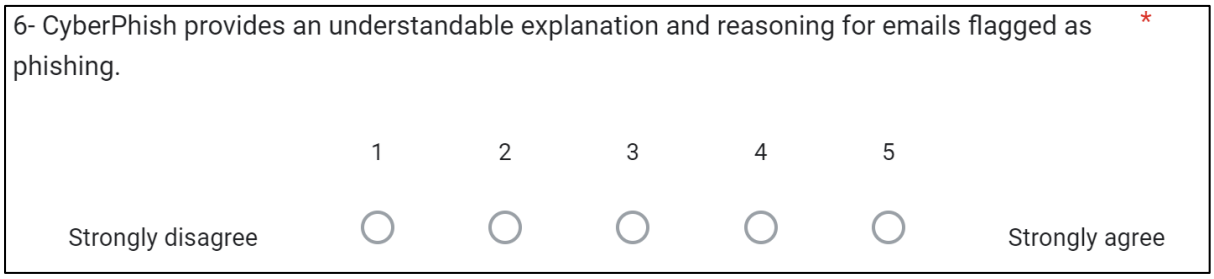


Figure 107: Question 6 in 10.2

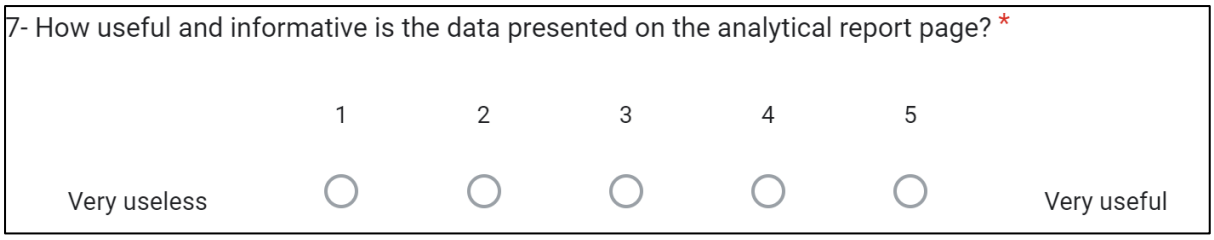


Figure 108: Question 7 in 10.2

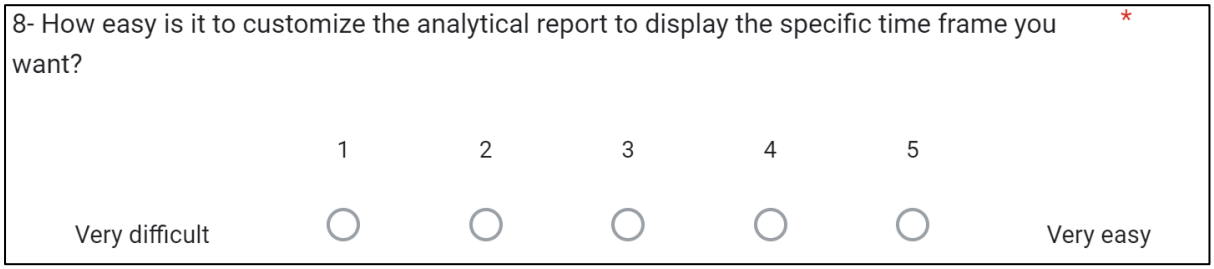


Figure 109: Question 8 in 10.2

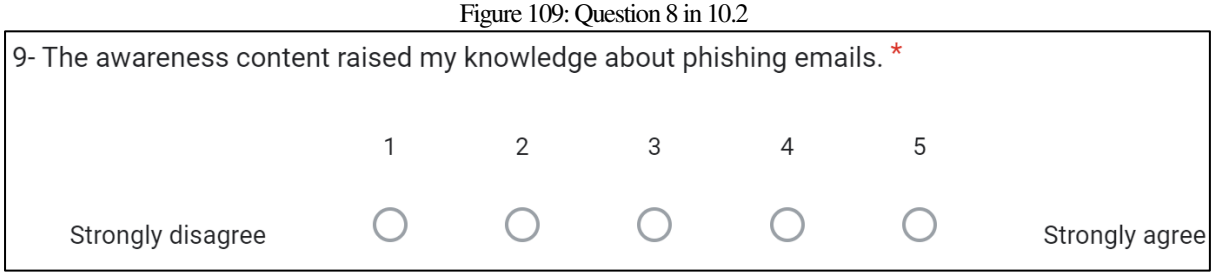


Figure 110: Question 9 in 10.2

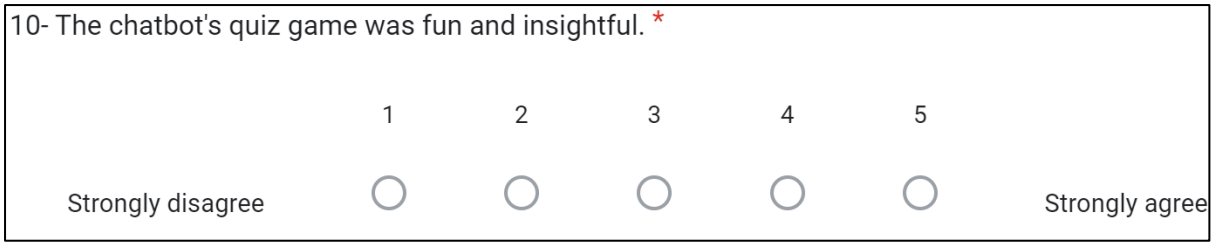


Figure 111: Question 10 in 10.2

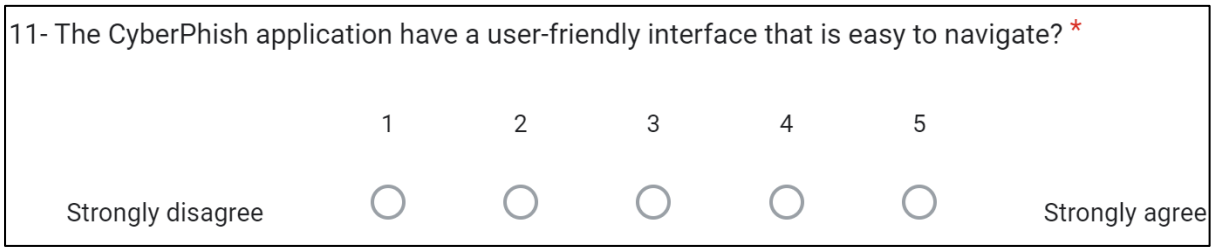


Figure 112 : Question 11 in 10.2

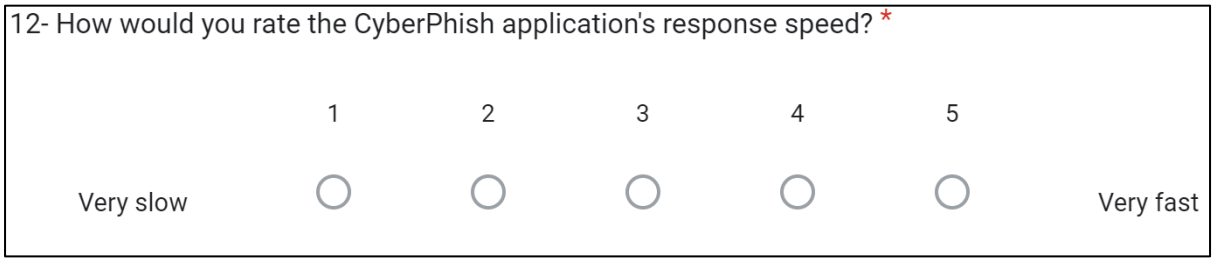


Figure 113: Question 12 in 10.2

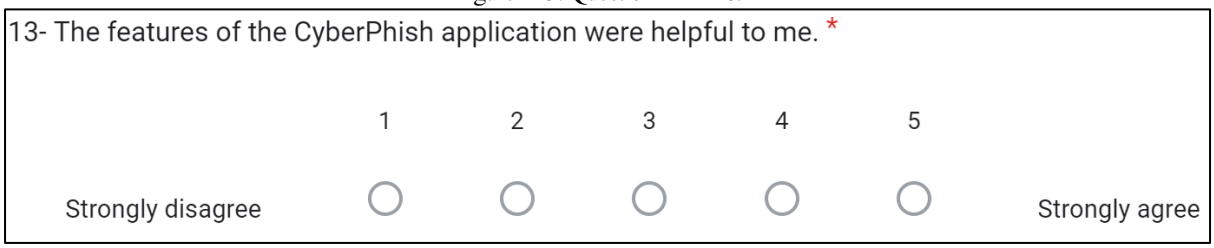


Figure 114: Question 13 in 10.2

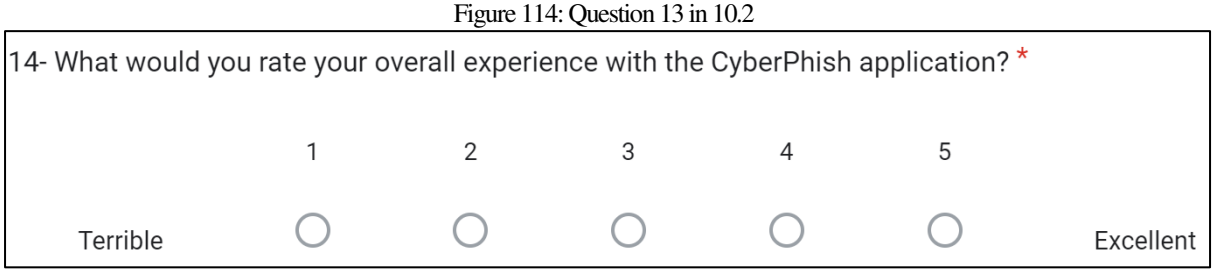


Figure 115: Question 14 in 10.2

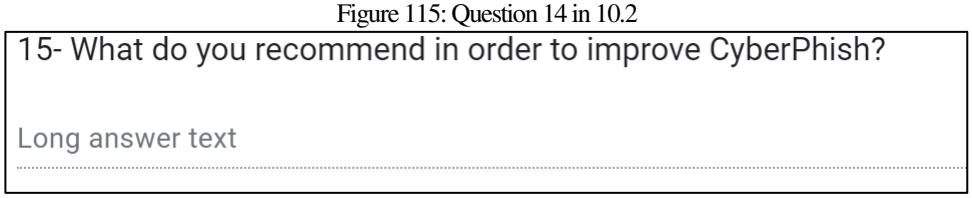


Figure 116: Question 15 in 10.2