

Developing a Fast Scam Prevention Mobile Application: Large Language Models

By

POON JIN YANG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONOURS)

COMMUNICATIONS AND NETWORKING

Faculty of Information and Communication Technology
(Kampar Campus)

FEBRUARY 2025

COPYRIGHT STATEMENT

© 2025 Poon Jin Yang. All rights reserved.

This Final Year Project report is submitted in partial fulfillment of the requirements for the degree of Bachelor of Information Technology (Honours) Communications and Networking at Universiti Tunku Abdul Rahman (UTAR). This Final Year Project report represents the work of the author, except where due acknowledgment has been made in the text. No part of this Final Year Project report may be reproduced, stored, or transmitted in any form or by any means, whether electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the author or UTAR, in accordance with UTAR's Intellectual Property Policy.

ACKNOWLEDGEMENTS

I wanted to say thank you to all of the people who has been and still is supporting me along the journey of researching and developing the project. I am grateful that my supervisor Dr Aun Yichiet has been very supportive of the direction of this project title and has been providing valuable feedback to further enhance the effectiveness of the project's solution in tackling the real world's problem. Additionally, I wanted to thank my parents who has been very supportive towards my study and has constantly ensure that I am in great shape. Throughout the journey of working on this project, I realised the importance of our body's well-being as being in a bad shape will definitely affect the quality of this project's research and report. I wanted to share that one key factor of the success of this project title is that the topic was picked by myself and I have devoted many days and night to ensure the final product of this project is up to my expectation and standard. Therefore, upon completion of this project's report, I hoped that this paper would be in great use of anyone who is proceeding in a similar direction for their project. Thank you very much, and I wish you luck on working with your research.

ABSTRACT

Scams are on the rise and constantly evolving. Threat actors have abused the rise of LLM to ease the process of creating deception information for scams. Manually flagging scam information is tedious and needs to be faster to counter the rapid growth of scam cases. Therefore, this project proposes developing a "*Sentinel*," a real-time scam detection system leveraging a large language model (LLM) for enhanced analysis of scam audio and text messages. The strategy is to build a mobile application that automatically captures the user's text and audio input and then utilizes Google's Gemini LLM for content analysis which maximizes the flexibility of the scam detector deal with new contents smoothly. After the complete LLM analysis is ready, the application will alert the user regarding the content analysis. The project considers the importance of the user's privacy by building an active application where user content analysis will only be done upon request, ensuring that the user has full control over when and how their data is analysed. The developed application would be capable of being implemented in the majority of Android devices with minimized performance hit on lower-end smartphones.

Area of Study: Cybersecurity, Artificial Intelligence

Keywords: Scam Detection, Large Language Models, Real-Time Analysis, Android Application, Privacy-Preserving AI

Table of Contents

COPYRIGHT STATEMENT	II
ACKNOWLEDGEMENTS.....	III
ABSTRACT	IV
Table of Contents.....	V
LIST OF FIGURES	VIII
LIST OF TABLES.....	XIII
CHAPTER 1.....	1
Introduction	1
1.1 Problem Statement and Motivation.....	3
1.2 Research Objectives	7
1.3 Project Scope and Direction	8
1.3.1 Project Scope.....	8
1.3.2 Project Direction	10
1.4 Contributions.....	11
1.5 Report Organization	12
CHAPTER 2.....	13
Literature Reviews	13
2.1 Previous Work on Fast Scam Prevention Mobile Application	13
2.1.1 Scam classification.....	13
2.1.2 Scam Detector Models	14
2.1.3 Existing Anti Scam Applications.....	16
2.1.4 Strengths and Weakness.....	27
CHAPTER 3 System Methodology/Approach	30
3.1 System Design Diagram	30
3.1.1 System Architecture Diagram	30

3.1.2 Use Case Diagram and Description.....	32
3.2 System Requirements.....	34
3.3 System Design.....	37
3.4 Project Milestones	38
3.4.1 Project 1 Milestones	38
3.4.2 Project 2 Milestones	39
3.5 Estimated Cost	40
3.6 Concluding Remark	40
CHAPTER 4 : System Design.....	41
4.1 System Block Diagram.....	41
4.2 System Components Specifications and Design.....	44
4.2.1 Converting screenshot to text.....	44
4.2.2 Google Chatbot Gemini.....	56
4.2.3 IP Quality Score	62
4.3 Sample System Components Interactions.....	63
CHAPTER 5 System Implementation	65
5.1 Hardware Setup and Configuration	65
5.2 Software Setup and Configuration	68
5.2.1 Android studio	68
5.2.2 Google AI studio	71
5.2.3 IP Quality Score	73
5.3 Integrating Services and Coding	75
5.3.1 Speech-To-Text.....	76
5.3.2 Gemini API Integration	79
5.3.3 IPQS API Integration.....	82
5.3.4 Google Machine Learning Kit Optical Character Recognition.....	85
5.3.5 Secure Browse	87

5.4 System Operation	93
5.4.1 Screen Scanning Activation Via Quick Access Panel	93
5.4.2 Screenshot Scanning Via Image Upload inside Application	95
5.4.3 Audio Scanning Activation Via Quick Access Panel.....	96
5.4.4 Safe Browsing Usage inside Application	98
5.4.5 Scan Quick Response (QR).....	100
5.4.6 Deleting Database Inside Application	101
5.5 Implementation Issues and Challenges	102
5.6 Concluding Remark	103
Chapter 6 System Evaluation and Discussion	104
6.1 System Testing and Performance Metrics	104
6.1.1 Load Testing.....	104
6.1.2 Response Time.....	105
6.1.3 Network Performance	110
6.1.4 Scam Detection Accuracy with Gemini LLM.....	114
6.2 Testing Setup and Result	118
6.3 Project Challenges	131
6.4 Objectives Evaluation	132
6.5 Concluding Remark	133
CHAPTER 7.....	134
7.1 Conclusion.....	134
7.2 Recommendation/Future Work.....	136
References	A1
POSTER	A9

LIST OF FIGURES

Figure Number	Title	Page
Figure 2.1	The Semakmule Portal	16
Figure 2.2	The Mobile Version of Semakmule Portal on Google Play	17
Figure 2.3	Details of the Check Scammers CCID	17
Figure 2.4	Truecaller Profile on Google Play	18
Figure 2.5	Truecaller Profile on Google Play Demonstration	19
Figure 2.6	Truecaller Auto Identifying Phone Number	19
Figure 2.7	Whoscall on Google Play	21
Figure 2.8	Whoscall App Info on Google Play	21
Figure 2.9	Scan Result of ScamAdviser	22
Figure 2.10	Positive and Negative Highlights of the Reviewed Website	23
Figure 3.1	High Level Overview Screen Scanning Feature	30
Figure 3.2	High Level Overview Audio Scanning Feature	31
Figure 3.3	High Level Overview Secure Browse Feature	31
Figure 3.4	Use Case Diagram Part One	32
Figure 3.5	Use Case Diagram Part Two	33
Figure 3.6	Project I Milestones	38
Figure 3.7	Project II Milestones	39
Figure 4.1	System Design of Screen Scanning Service	41
Figure 4.2	System Design of Audio Scanning Service	41
Figure 4.3	System Design of Secure Browse Service	42
Figure 4.4	Google Recognition Model Recognizing Text	49
Figure 4.5	Google Guide Example OCR Image	50
Figure 4.6	Google Guide Example OCR Result part 1/3	50
Figure 4.7	Google Guide Example OCR Result part 2/3	51
Figure 4.8	Google Guide Example OCR Result part 3/3	51
Figure 4.9	The Current Smartphone Screen	52
Figure 4.10	System Log Capture Screenshot Success	53
Figure 4.11	Google Machine Learning OCR Result	53
Figure 4.12	CPU and Memory Usage Text Conversion Process	54
Figure 4.13	Detail CPU and Memory Usage Text Conversion Process	54
Figure 4.14	Transformer Model from Original Paper with Additional Comments	58

Figure 4.15	Tokenizing Words	60
Figure 4.16	Sample Picture	63
Figure 4.17	Gemini Result	63
Figure 4.18	URL Found	64
Figure 4.19	IPQS Scan Result	64
Figure 5.1	Image of Xiaomi Poco X3 NFC	65
Figure 5.2	Phone Configuration Part I	66
Figure 5.3	Phone Configuration Part II	66
Figure 5.4	Phone Configuration Part III	66
Figure 5.5	Phone Configuration Part IV	67
Figure 5.6	Phone Configuration Part V	67
Figure 5.7	Phone Configuration Part VI	67
Figure 5.8	MSI GF63 Thin 10UC	67
Figure 5.9	Android Studio Download Page	68
Figure 5.10	Android Studio Android Manifest XML File	68
Figure 5.11	Android Studio Build Grandle Kotlin File Dependencies Part I	69
Figure 5.12	Android Studio Build Grandle Kotlin File Dependencies Part II	69
Figure 5.13	API Keys Properties File	70
Figure 5.14	Android Studio Setting Up API Keys	70
Figure 5.15	Google AI Studio Webpage	71
Figure 5.16	Google AI Studio Navigate to API Key Page	71
Figure 5.17	Google AI Studio Create API Key	72
Figure 5.18	Google AI Studio Retrieve Gemini Key	72
Figure 5.19	IPQS Login Page	73
Figure 5.20	IPQS Registration Page	73
Figure 5.21	IPQS Navigating to API Key	74
Figure 5.22	Audio Capture in Process	77
Figure 5.23	Gemini Scanning the Conversation	77
Figure 5.24	Gemini Result	77
Figure 5.25	Gemini API Model Code Configuration	79
Figure 5.26	Gemini Instruction Part I	80
Figure 5.27	Gemini Instruction Part II	80
Figure 5.28	Gemini Result Implementation	81
Figure 5.29	IPQS Calling API Code	82

Figure 5.30	IPQS Analyzing the JSON Response	83
Figure 5.31	IPQS Main Function	84
Figure 5.32	Google ML Kit OCR Callback Method	85
Figure 5.33	Google ML Kit OCR Analyze Method Variant I	85
Figure 5.34	Google ML Kit OCR Analyze Method Variant II	86
Figure 5.35	Secure Browse Scan and Load URL	88
Figure 5.36	Secure Browse Scanning URL Code	88
Figure 5.37	Secure Browse Gemini API Scanning	89
Figure 5.38	Secure Browse Secure Web View Client	89
Figure 5.39	Secure Browse Update User Interface	90
Figure 5.40	Secure Browse Perform URL In Depth Part I	90
Figure 5.41	Secure Browse Perform URL In Depth Part II	91
Figure 5.42	Opening Quick Access Panel	93
Figure 5.43	Screen Scanning Activation Via Quick Access Panel	93
Figure 5.44	Screen Scanning Allow Application Screen Permission	93
Figure 5.45	Screen Scanning Edit Page	94
Figure 5.46	Screen Scanning Pending Result	94
Figure 5.47	Screen Scanning Result Part I	94
Figure 5.48	Screen Scanning Result Part II	94
Figure 5.49	Screen Scanning Activation Inside Application	95
Figure 5.50	Upload Image to Screen Scanning	95
Figure 5.51	Screen Scan Analysing	95
Figure 5.52	Screen Scanning Result Part I	95
Figure 5.53	Screen Scanning Result Part II	95
Figure 5.54	YouTube Scam Voicemail Example	96
Figure 5.55	Audio Scanning Activation at Quick Access Panel	97
Figure 5.56	Audio Scanning Window	97
Figure 5.57	Audio Scanning Sound Listened	97
Figure 5.58	Audio Scanning Alert	97
Figure 5.59	Secure Browse Activation	98
Figure 5.60	Secure Browse Page	98
Figure 5.61	Secure Browse Analyzing	98
Figure 5.62	Secure Browse Threat Result	98
Figure 5.63	Secure Browse Scanning YouTube	98
Figure 5.64	Secure Browse Safe Result	98
Figure 5.65	Secure Browse Load Website With No Javascript	99

Figure 5.66	QR Sample	100
Figure 5.67	Scanning QR	100
Figure 5.68	QR Load To Secure Browse	100
Figure 5.69	Scanning Complete Result Safe	100
Figure 5.70	Secure Browse Loading Website	100
Figure 5.71	Accessing Scanned URL Database	101
Figure 5.72	Navigating to URL Database	101
Figure 5.73	Selecting All URL	101
Figure 5.74	Deletion of All URL	101
Figure 6.1	Overall CPU and Memory Usage Running Developed Application	104
Figure 6.2	Peak Resource Usage Running Developed Application	104
Figure 6.3	Code Integrated for Logging Timestamp	105
Figure 6.4	Screen Scan Test I	106
Figure 6.5	Screen Scan Test II	106
Figure 6.6	Screen Scan Test III	106
Figure 6.7	Screen Scan Test IV	106
Figure 6.8	Screen Scan Test V	107
Figure 6.9	Timestamp Log for Each Test	107
Figure 6.10	Audio Scanning Timestamp Log Code	108
Figure 6.11	Audio Scanning Timestamp Log	109
Figure 6.12	Gemini Timestamp Logging Code	110
Figure 6.13	IPQS Timestamp Logging Code	110
Figure 6.14	Gemini Timestamp Log Part I	110
Figure 6.15	Gemini Timestamp Log Part II	111
Figure 6.16	Gemini Timestamp Log Part III	111
Figure 6.17	IPQS Test Case	112
Figure 6.18	IPQS Timestamp Log	112
Figure 6.19	Confusion Matrix of Gemini with 100 Dataset	116
Figure 6.20	Number Scores of Gemini with 100 Dataset	117
Figure 6.21	Testing Task 1	118
Figure 6.22	Testing Task 2, 3, 4	119
Figure 6.23	Message Received I	119
Figure 6.24	Message Received II	120
Figure 6.25	Message Received III	120
Figure 6.26	Building Decoy	121

Figure 6.27	Received Message 3 Inside Link	121
Figure 6.28	Performance Task 2, 3, 4 without Application Assistance	122
Figure 6.29	Performance Task 2, 3, 4 with Application Assistance	122
Figure 6.30	Testing Task 5, 6, 7	123
Figure 6.31	Purchase Source 1 Fake JBL Website	123
Figure 6.32	Purchase Source 2 Lazada Website	124
Figure 6.33	Purchase Source 3 Shopee Website	124
Figure 6.34	Virus Total Scan	125
Figure 6.35	Figure 6.35 Performance Task 5, 6, 7 without Application Assistance	126
Figure 6.36	Figure 6.36 Performance Task 5, 6, 7 with Application Assistance	126
Figure 6.37	Audio File 1	128
Figure 6.38	Audio File 2	129
Figure 6.39	Performance Task 8, 9 without Application Assistance	129
Figure 6.40	Performance Task 8, 9 with Application Assistance	130

LIST OF TABLES

Table Number	Title	Page
Table 2.1	Summary of Previous Work Done	24
Table 3.1	List of Software and Programs Used for Development	34
Table 3.2	Specifications of Laptop	36
Table 3.3	Specifications of Smartphone	36
Table 5.1	Phone Configuration	66
		67
Table 5.2	Secure Browse Methods and Description	91
		92
Table 6.1	Response Time Screen Scanning	108
Table 6.2	Response Time Audio Scanning	109
Table 6.3	Network Performance Gemini API	111
Table 6.4	Network Performance IPQS API	113

CHAPTER 1

Introduction

In 2022, the National Scam Response Centre (NSRC) Malaysia had received over fifteen-thousand reports of scam cases which has caused RM141 million losses from the victims ever since the launch of NSRC entity in October 2022 [1]. In addition, the number of unique phishing sites according to Anti-Phishing Working Group (APWG) has reported to be 1,624,144 sites at the first quarter of 2023 [2]. Based on these reports, it could not be ignored the fact that phishing and scamming activities are on the rise globally. If we had chosen to neglect this issue, then the individuals in the society will suffer major impact on their safety and finance. If law enforcement or the government fails to tackle this issue, then the reputation of the country will be greatly affected.

Nowadays, as our society is embracing ourselves into a fully digital world such as digital economy, work and education, artificial intelligence (AI) and automation, smart infrastructure, etc. where the amount of technology involved in our lives has greatly increased. This has significantly ease everyone in completing their own task and has replaced simple repetitive task with machines or systems that are able to do it faster and more efficiently than us humans.

Without a doubt, the rise of technology has greatly benefited the society, but this comes at a cost. The rise of technology not only ease everyone in performing normal task, but also allowed threat actors to utilize those technologies to further take their crimes to the next level. Threat actors are able to utilize the current technology to reach out more people to become their victim. Additionally, with the recent rise of the Large Language Model (LLM), the threat actors are reported to be using the LLMs to perform malicious act such as influence operations [3].

When Internet has become more popular around the globe, it is concluded that this technology was not only taking those traditional crimes such as property theft, information theft, etc. to online but it has also introduced a series of dangerous crimes which includes but not limited to hacking, botnets, bots to troll victims on the Internet [4]. Keep in mind that the rise of the Internet has been evolutionary for the entire human society during that era while during the time of writing this report the rise of LLMs are currently happening. Therefore, the problem was never “will the rise of LLM or Artificial Intelligence (AI) introduce a new series of dangerous

crimes” but rather it has become a matter of time that such unfortunate event happens. One of the examples that shows threat actors started to utilize LLM is an article written by Elgan who has talked about a special version of AI module based on GPTJ, an open-source large language model, called WormGPT [5] [6]. The WormGPT was a smart tool for the threat actors which it was capable of producing content that would spread disinformation and possibly assist the threat actors in producing phishing emails, scamming or anything relevant [7].

On the other hand, over half of the global population, as the time of the report is written, now owns a smartphone, which means over half of the global population has become a potential target of cybercriminals [8]. In 2022, smartphones have been predicted to become a more focused target for the cybercriminals due to the device have instant access to email, the internet, and many other applications. Meanwhile it does not have any advance security measures such as firewalls or third-party anti-virus application unlike our personal computers are equipped with [9].

Hence, it has raised the idea of building a mobile application with AI chatbot to solve the rise of scam cases rising. The implementation of the LLM chatbot in the application is used to understand the context or conversation and detect disinformation that would harm the user or scam the user. This project shows a potential solution in reducing the scam cases without involving a tremendous amount of money in a long-term operation. One great feature that this application could provide is the ability to learn and adapt to the latest scamming trend with the least amount of human assistance possible while also being fast and responsive to use. Therefore, due to its learning ability, this application could be built to be future-proof and the accuracy of the application’s scam detection could only improve over time.

1.1 Problem Statement and Motivation

In order to counter cybercriminals scamming people or falling into their malicious act, the people in the society must keep themselves updated with the latest information of the cybercrimes currently occurring and targeting the society to alert themselves and find ways to prevent falling victim of these malicious acts. However, we have to account for the information to reach everyone before the cybercriminal reach them.

Hence, it will rely on the ability of the individuals to retrieve the latest preventive measures to protect themselves from online threats. Additionally, the individuals will also need to differentiate the false information spreading online and only listen to those information from legit parties such as the official Avast anti-virus software social media account instead of random users or false advertisement that includes misleading information online. According to research from three MIT scholars, they have concluded that false news spreads more rapidly on social network where they have tested on Twitter (known as X by the time of writing this report) [10], which means it is very crucial that online users to develop skills to be able to recognize false information on the Internet.

Unfortunately, not everyone has the time or the will to equip themselves with the latest preventive measures to protect themselves against cybercriminals. As a result, they will face the consequences of their lack of awareness of the current threats lurking online. Their consequences could be ransomware, private information leaked, bank information stolen, etc. Additionally, majority of the people in the society are emotional and some of them may be easily influenced by their emotions to perform certain behaviors or actions.

This characteristic is not necessary a negative feature of the human being as it is the primary factor that led us to help each other out when crisis happens; however, one can abuse this behavior to perform malicious actions that can cause massive impact to the victim. It can be concluded to two primary factors that has caused the people in the society to fall victims to the cybercriminals:

a. Lack of Vigilance

People in the society are busy with their lifestyle and daily chores. Their daily life may be fully occupied with tasks and do not consider allocating time to equipped themselves with the latest knowledge about the acts of cybercriminals and the disinformation that is spreading online. This would result them believing false information and fall into the malicious scheme of the cybercriminals.

b. Emotional Triggers

Humans are emotional creatures, which those emotions are able to affect how we respond to the situation that we are currently facing [11]. Unlike machines, the influence of our own emotions could lead to inconsistent behavior and reaction when we encounter similar situation. For example, normally a person who receives a one-time password (OTP) from their messaging application on his/her smartphone, he/she will instantly be alerted and sense danger that another party is attempting to steal their money. However, if the cybercriminal had called the victim with a rushing tone and attempts to make the victim to believe he is an old friend, then there exists a chance that the victim will give the OTP number to the threat actor although normally he/she wouldn't. If one person has been influenced by another person which makes them be in a rushing and panicking mode, their thoughts may not be rational enough to be aware that they are currently falling into a malicious scheme.

Motivation

The rise of Artificial Intelligence (AI) is currently happening, particularly the rise of Large Language Model (LLM). Although it is capable of assisting the threat actors in their malicious activities, we can also utilize LLM to counter their malicious schemes. The simplest threat that can be spotted online is false information which can be separated into many categories such as phishing emails, false advertising, misleading information, etc. Hence, with the help of LLM that can understand paragraphs of words and human conversations, we could potentially utilize this technology to spot and alert the user about the danger that they are currently facing.

As smartphones are a common device owned by the people in the society, we could build a mobile application that is equipped with capability that allows it to ensure what the user received or is currently viewing is safe to proceed. This idea would potentially reduce the number of people becoming victims of scams and disinformation if majority of the people utilized the application.

The design of the system can be made so that the LLM would learn from all of the user to spot malicious activities that they are currently facing. This would allow it to quickly update its knowledge about the trending malicious scheme that the cybercriminals are currently using. This would solve the first problem statement proposed which is the lack of vigilance as now the application would assist in alerting the users about potential dangers that they might have overlooked. Additionally, the application is powered by an AI which has no emotion, and their outputs are more consistent than humans.

It may be argued that the user could visit the web interface of the LLM that was provided by the LLM companies such as chatgpt.com and gemini.google.com. However, on the smartphone, when the user has encounter information that he/she is unclear and unsure, he/she may proceed to directly search for the information on the Internet instead of checking with one of the LLM. This would still have a chance that the user would encounter another false information spreading online as the search engine is not designed for detecting any disinformation but instead the primary purpose of the search engine for example Google's search engine is to explore the web regularly to find pages to add to their index. Additionally, to have the users to separately and manually search for the information that they are

CHAPTER 1

encountering would require the users to have some degree of suspicion of the content that they are reading. This is because the users are just unsure of the content and does not have any suspicion on the content, they may not want to deal with the hassle to load the web application to verify the content. Therefore, the users are still facing the same problem that was proposed previously, the lack of vigilance.

That being the case, this is one of the benefits of the application because if the mobile application proposes a way that the users would only need to click one button and it will scan the current mobile screen of the users then verifying the information in front of them, this would no longer be a hassle for them and would instead encourage the users to constantly verify the content that they are viewing. This would effectively solve both problem statement that was proposed by this report which are “lack of vigilance” and “emotional triggers”.

Aside from that, the mobile application can make it so that it would only need one click of button so that it would open the microphone to listen to the audio that the user desires to for scanning the content purpose. This would also allow the user to listen to any other audio source such as his/her personal computer which is playing the audio of a video from Facebook, a social media platform.

1.2 Research Objectives

1. To develop a privacy focused scam prevention and detection method to mitigate scam on smartphones

The application developed will be built to mitigate scams on smartphones and also take into consideration about the user's privacy. The application will achieve "mitigating scam on smartphones" by using LLM to analyze the input and alert the user. Then to achieve "protecting user's privacy" the application would only run upon user's request instead of running in the background 24/7.

2. To develop a scam message detection using LLM

To effectively develop a scam message detector, the detector is equipped with LLM to analyze the input content. The LLM service will be responsible for flagging the content to be safe or dangerous. One of the advantages of implementing existing LLM services that are provided by third-party as the scam detector is the maintenance of the scam detector would be the responsible of the LLM developer company.

3. To develop scam audio detection using audio description and LLM

To increase the coverage of the scam detector, the mobile application will be built with audio capture ability as an alternative input capturing channel. This allows the overall coverage of the mobile application to detect scams over text and audio.

1.3 Project Scope and Direction

1.3.1 Project Scope

The proposed solution of the project is to build a mobile application that would be used by anyone to prevent scam by detecting disinformation. Hence, protecting the user from malicious intent of threat actors. To reduce the size of the project to be completed within the determined period of time, the project will introduce its scope.

Firstly, the scope of the project is limited to Android environment only, which means the mobile application is not built for Apple's mobile operating system or any other mobile operating system. Secondly, the application built will consist characteristics such as fast reacting, accurate detection, and convenient to use. Further details on the characteristics are explained below with bullet points:

a. Fast Reacting

The mobile application is expected to be capable of producing the result of the scan from the input information within a short period of time. The length of the time that would be considered "short" in this project is more or less than 5 seconds of reaction time for the mobile application. This will ensure the mobile application will avoid making the user to feel the application is unresponsive.

b. Accurate Detection

The scan result of the mobile application should be ensured to reach 80% or more for its accuracy. It is to gain trustworthiness of the mobile application from the users. If the accuracy of the detection is low, it would fail to meet the goal of this project.

c. Convenient to Use

The mobile application should be built with minimal human interaction in order to make it easy to use. This is crucial as building an application with very easy-to-use user interface would gain the liking of the users and potentially make them rely on the application as it is very easy to operate. As a result, more people would be able to verify their information or at least the chances of them being a victim of disinformation would greatly be reduced.

CHAPTER 1

Aside from that, another limitation of the project is the scalability of the users that it can serve. The outcome of the project would prove the point that the idea of this project title “Developing a Fast Scam Prevention Mobile Application: Large Language Models” is possible, which means the project not considering to be building a robust system that is suitable for serving massive number of users. However, this can be solved by building a centralized backend system hosted on an online cloud, scalability is not being prioritized in this project, but it does not mean the system could not be modified to suit the desired requirement.

In addition, the mobile application would be only accepting text and audio input only and will not be considering sending content that are not text, for example, a picture of a cat, to the system to analyse. The audio is accepted as an input for the application because the application would be converting the audio to text and only then it will be processed.

The application system will also only be working with one language which is English as it is the international language of the world. However, modification of the application to suit other languages are possible but implementation of such configuration is not prioritized in this project.

1.3.2 Project Direction

The project direction is to build a mobile application that prevents scams. It achieves this goal by detecting disinformation on the user's current smartphone screen. The application will retrieve the text on the screen and process it by sending it to LLM for analysis and a third-party website link scanner.

The application is built with the mindset of being suitable for anyone, anywhere, and anytime, to use for detecting disinformation and protecting themselves from scams as long as the user has access to the Internet. One of the proposed problem statements is "Emotional Triggers", in which the user's mood and emotion could affect their behaviour, increasing their chances of falling victim to cybercriminals. We should also consider that age also plays a crucial factor in affecting an individual's awareness. According to a study conducted by a group of professors, in short, they found out that there exists a positive relationship with the elderly who have certain characteristics to the susceptibility to scams [12]. They found out that susceptibility to scams was positively correlated with the elderly's age, frailty, ADL and IADL disability, neuroticism, and loneliness. Additionally, this test was conducted without elderlies with dementia, and one can only expect their judgement to prevent themselves from landing in the hands of cybercriminals to be poorer than those without dementia.

On the other hand, youngsters are also vulnerable to online scams, according to the Federal Trade Commission. Scams affect all age groups, but the rate of people under the age of 20 is unexpectedly highest among all groups [13]. It is safe to say that no particular age group is protected from scams or disinformation on the Internet. They could only blame it on themselves with the lack of vigilance and knowledge once they had become the victim of the cybercriminals.

Hence, this is where this project aims to propose a solution to counter it. With LLM like ChatGPT and Gemini, as well as other services such as IP quality score (IPQS) to scan malicious website links combined and working together in an application, we can protect people in society from cyber threats lurking on the Internet.

1.4 Contributions

The contribution of this project towards society aims to provide a trustworthy, convenient, and effective solution to counter the disinformation circulating over the Internet and help to prevent scams happening to the people in the society. The following below are the three main features that the application will provide to its users:

a. Quick Response time

The mobile application design will be powered by a LLM and third-party website link scanner to prevent scams by detecting disinformation or malicious content. The delay of response from the Application Programming Interface (API), which will be used to implement the LLM and third-party services into the application, is tolerable and arguably fast. A testing on the response time from Google's LLM, Gemini, API has been able to response within 10 seconds and would be shorter if the length of the prompt is shorter. Therefore, the system is "quick".

b. Efficient Automated Scanning System

The mobile application is promoted to be easy to use, the application design achieves this by having minimal human interaction needed for the entire process of verifying the input content. The input content can be audio or text; however, it will not require the user to upload a audio file or manually input the text into a input column for analysis. The application would be designed to have a button that when clicked it will just scan the current screen of the user's smartphone and scan the text. Or if the user click on the listen function button then it will actively listen to the surrounding sound and pick the closes audio to analyse. Hence, it will promote a automated and efficient scanning system.

c. High Availability

The system would be build in one of the largest mobile operating system to be made available for a large portion of the population to install on their smartphone.

1.5 Report Organization

This report will be separated into 7 chapters to neatly organize the research and development conducted when developing the anti-scam mobile application. In Chapter 1, it includes the goal, objectives, and motivation to start this project. In Chapter 2, it will include the literature review on previous work or similar work done on a similar topic. Then in Chapter 3 which will include the high-level overview of the system design, project milestones, and estimated cost. Chapter 4 will be explaining the system design in lower-level by providing more technical details of the application development. Chapter 5 will be including the hardware and software setup, code implementation, and system operation showcase. Chapter 6 will include testing and performance. Finally, Chapter 7 will conclude the entire project.

CHAPTER 2

Literature Reviews

2.1 Previous Work on Fast Scam Prevention Mobile Application

2.1.1 Scam classification

The ever-increasing number of scam cases occurring around the globe can be overwhelming to keep track, as more scam types are on the rise to seek opportunities that aims to exploit vulnerable individuals. Based on the proceedings of the Cyber Secure Nigeria Conference (CSEAN) in 2023, they had collected 419 samples of scam emails from the year 2023. In short, the 419 scam emails were classified into 27 unique types of scams. The 27 types of scams are Lucky winner scams, threat of exposure scams, Business/Partnership proposal scams, Supply scams, Dead or alive scams, Compensation scams, COVID-related scams, Next-of-kin scams, order scams, malware/ransomware scams, investment scams, cancer/long-term illness scams, money-in-box scams, loan scams, killed-father scams, fund scams, cryptocurrency scams, fake help scams, authorization scams, foreign affairs scams, contacts for marketing scams, spiritual scams, delivery scams, job offer scams, details scams, romance scams, and software development scams [14]. However, these scams are limited to email scam types and the scam emails analysed by the researchers were only collected from Nigeria instead of multiple countries.

Another source of scam types is from the Singapore Police Force; according to them, they have disclosed the topmost common scams that occur in Singapore, which are phishing scams, job scams, e-commerce scams, investment scams, fake friend call scams, and many other less frequent occurring types of scams. It should be noted that their analysis of the scam cases was based on the scam cases report in 2022, and the number of scam cases analysed by them was 31,728 number of cases [15]. The result of the scam classification has concluded that overall, the scam cases aim to abuse the trust of the victim without them knowing, which resulted in the scam victims suffering financial losses.

2.1.2 Scam Detector Models

As the number of scam cases rises, security researches around the globe has contributed to tackling the problem by proposing various solutions. One type of the solutions proposed is to have an affective scam detection model. Detection models proposed recent to 2024 are mostly related to Artificial Intelligence (AI) due to their capabilities and effectiveness in detecting scams. One of the examples is detecting telephone-based social engineering attacks using scam signatures by Derakhshan et al [16]. Their proposed method aims to effectively detect social engineering attacks specifically telephone-based scams. They had used scam signatures, which has been commonly used in antivirus to identify malwares with their malware signatures. Using scam signatures, they aim to effectively identify the key characteristics of the scam calls and prevent such attacks. They had used word embeddings and sentence embeddings which was a type of Natural Language Processing (NLP) tasks. Their method had resulted in accuracies for all scam types identify by them to be higher than 90%. It is worth mentioning that when their detection model has identified the that the conversation is a scam, it was guaranteed to be a scam which proposed no false positives in their detection model. In short, their proposed detection model has successfully showed its effectiveness in detecting scams.

Another researcher team Roy et al. has proposed to use a machine learning model (ML) to identify phishing prompts effectively. Their ML model is an advanced Natural Language Processing (NLP) model based on transformer architecture, RoBERTa model [17]. The RoBERTa stands for Robustly optimized Bidirectional Encoder Representations from Transformers (BERT) approach. Their study has tested of four types of phishing category detections on their machine learning (ML) model, which are individual prompt detection, phishing collection detection, phishing prompt subset detection in Real-Time, and detection of phishing email prompts. Their measurement unit used to determine the accuracy of the detection is the F-score or also known as the F1 score. Their ML detection model achieved an F1 score of not lower than 0.9 for all four types of phishing categories, which indicates that identifying phishing attacks effectively is feasible using ML models. Overall, their method of identifying the scams is by understanding the conversation where they had proposed four types of approaches related to their detection models.

2.1.3 Existing Anti Scam Applications

To tackle with the rise of scam cases in Malaysia, Malaysia Government has taken action to assist in solving the problem. According to The New Straits Times, during 2020 the Commercial Crime Investigation Department (CCID) Malaysia has launched the “Semakmule Portal” which is capable of verifying the status of bank accounts and phone number [18].

semakmule.rmp.gov.my

Jabatan Siasatan Jenayah Komersil
Polis Diraja Malaysia
Commercial Crime Investigation Department
Royal Malaysia Police

About Vision & Mission Contact us

Semak Akaun Yang Ada Report
English Bahasa Malaysia 中文 ភាសាខ្មែរ

Carian Telah Dibuat: 20,985,731 Carian [BSS 49,635]

Masukkan Nama Syarikat Penipuan: Masukkan Kata Kunci

Kategori: Nama Syarikat Penipuan
Akaun Bank
Nombor Telefon
Nama Syarikat Penipuan

Captcha: (umat Captcha)

Borang Soal Selidik Semak Maklumat

PENAFIAN
Kerajaan Malaysia dan PDRM tidak bertanggungjawab di atas kehilangan atau kerosakan disebabkan penggunaan mana-mana maklumat yang diperolehi daripada laman web ini.
Ia tidak memberikan hak kepada mana-mana pihak untuk bertindak bagi PDRM dalam apa-apa jua urusan.

Copyright Registration
LY2017001987 21 JUN 2017

Kumpulan Inovasi PDRM Digital API Saver

Muat turun Aplikasi ke telefon bimbit
Check Scammers CCID dari Google play

12 Bulan Whoscall Premium Tebus Dengan PERCUMA*
Kod Promosi: PDRMLawanScams
Di redeem.whoscall.com Sebelum 31/03/2024

Figure 2.1 The Semakmule Portal

Based on Figure 2.1 we can see that during the time of writing this proposal, the Semakmule Portal has offered ability to verify the name of the company. In addition, they had also released their own mobile version of Semakmule Portal named “Check Scammers CCID”, which was released on 2020 February 23 [19].

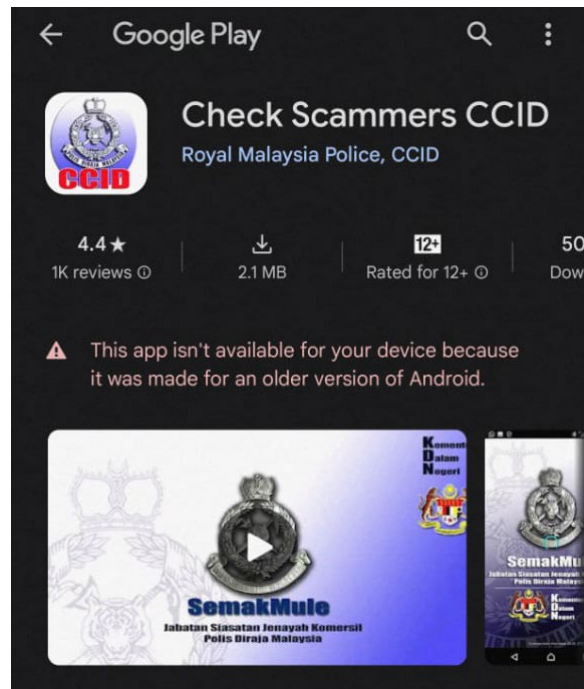


Figure 2.2 The Mobile Version of Semakmule Portal on Google Play

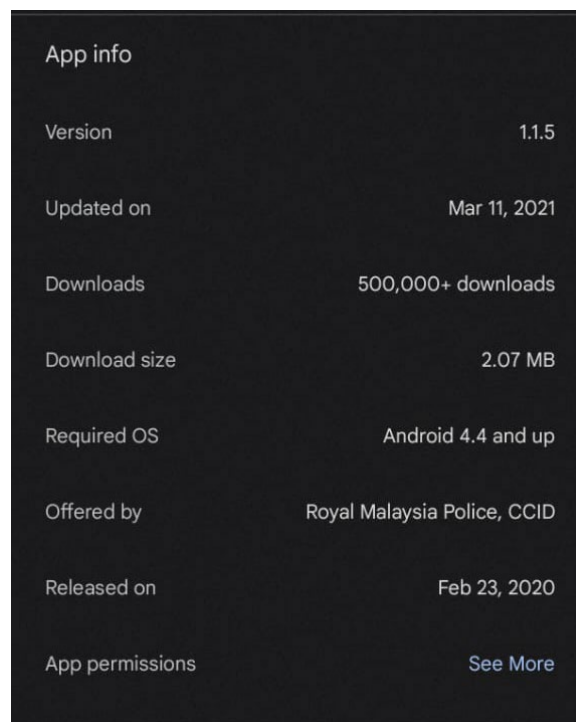


Figure 2.3 Details of the Check Scammers CCID

Based on Figure 2.3 we can see that the mobile application was released on 2020 February 23 and last updated on 2021 March 11 by the Royal Malaysia Police, CCID.

CHAPTER 2

Another anti scam application that was popular among users was called Truecaller. Truecaller is a mobile application origin from Sweden that aims to assist in blocking phone numbers of spam [20] [21]. The application utilizes the users' own contacts list and its Truecaller users' community reports to identify who is calling and whether it is a spam.

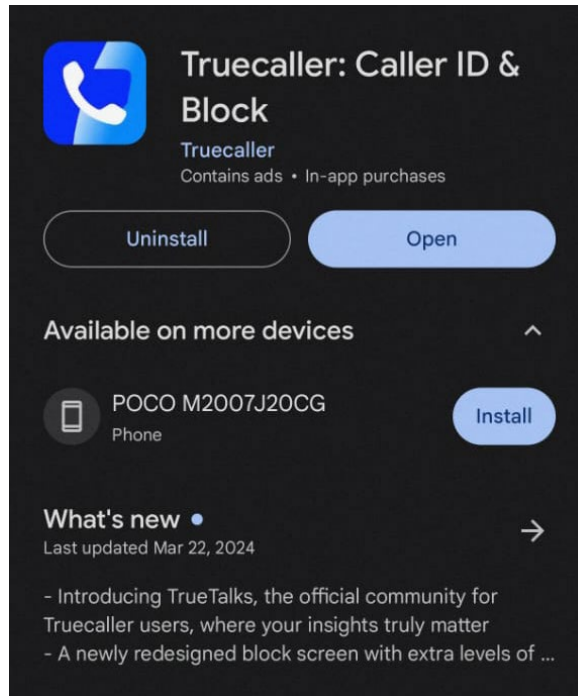


Figure 2.4 Truecaller Profile on Google Play

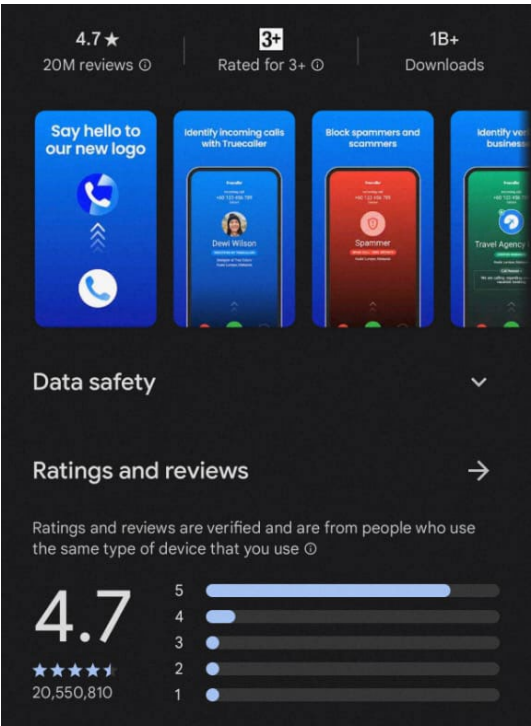


Figure 2.5 Truecaller Profile on Google Play Demonstration

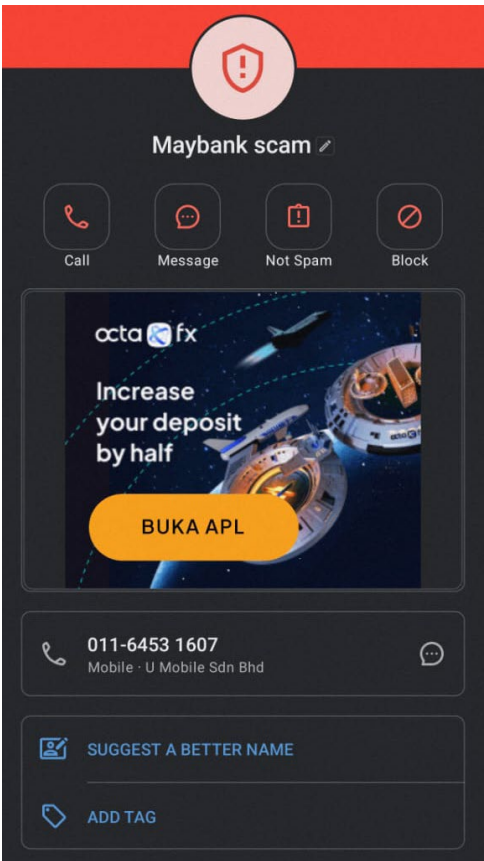


Figure 2.6 Truecaller Auto Identifying Phone Number

CHAPTER 2

Based on Figure 2.6, we can see Truecaller in action to identify the phone number without the phone user manually setting it. The application is able to share the reports of the community to other users to alert about potential spam or scam phone number. In addition, on September of 2023, Truecaller released a new feature called “Search Context” which was part of the Truecaller’s AI identity engine. The feature will allow the user to know whether the name assigned to the phone has been frequently or recently changed [22].

Another competitive mobile application that has similar features to Truecaller is “whoscall” where it aims to provide services that identifies unknown calls in real-time and filters out the spam calls via its own AI-powered system and large database [23]. The application has also established collaboration with Gogolook, a Taiwan company expert in anti-fraud technology, which offers their databases to the application for unknown callers detection and malicious calls prevention [24].

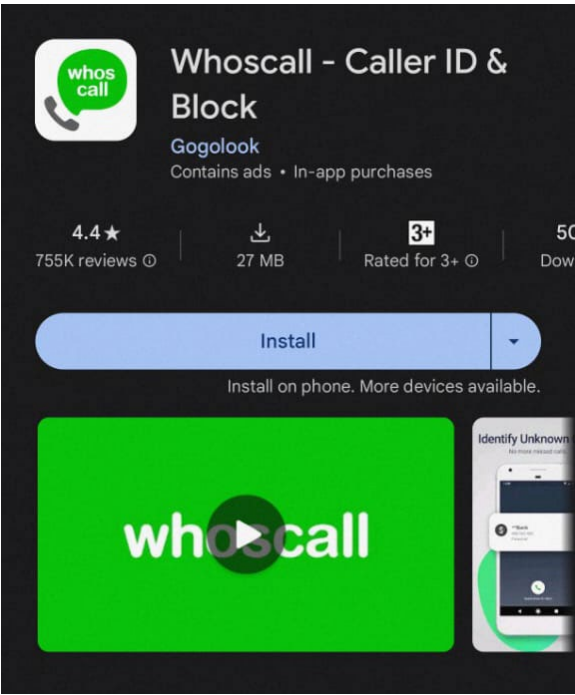


Figure 2.7 Whoscall on Google Play

App info	
Version	7.56
Updated on	Mar 14, 2024
Downloads	50,000,000+ downloads
Download size	27.00 MB
Required OS	Android 8.0 and up
In-app purchases	RM 3.50 - RM 449.99 per item
Offered by	Gogolook
Released on	Jul 24, 2010
App permissions	See More

Figure 2.8 Whoscall App Info on Google Play

ScamAdviser

ScamAdviser is an organisation that offers services to discover potential scam from malicious websites [25]. Their services was mainly provided to their customers via web applications back in 2012; however, as of current date in 2024 they have provided options to download their mobile application to utilize their services. They primarily help the user to check the validity of the website links, they do this by taking website links as input and checking information such as the age of the Uniform Resource Locator (URL), Internet Protocol (IP) address of the webserver, the availability of the contact details and ratings to the review sites, and much more that they don't disclose [25].

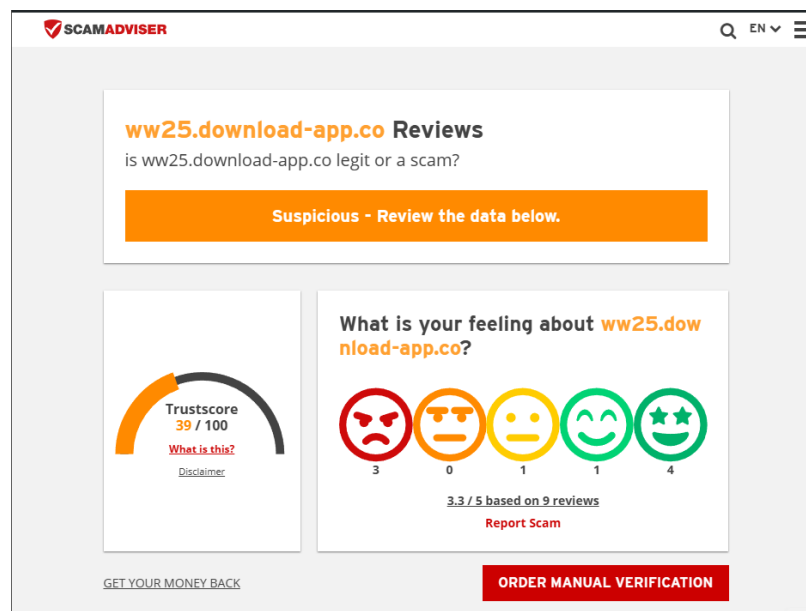


Figure 2.9 Scan Result of ScamAdviser

The result of the ScamAdviser when its scan is completed for a website will show a score from 1 to 100 according to the trustworthiness of the website (as shown in Figure 2.9), the higher the score the higher the chance of the website is safe and reliable. The result can also show negative indicators that will decrease the score of the website, those indicators include high-risk country, hiding or disguising ownership details, or even using a new domain. One great feature of their score is that the score will not be fixed and will change over time when new data is gathered related to the websites that it reviews [26].

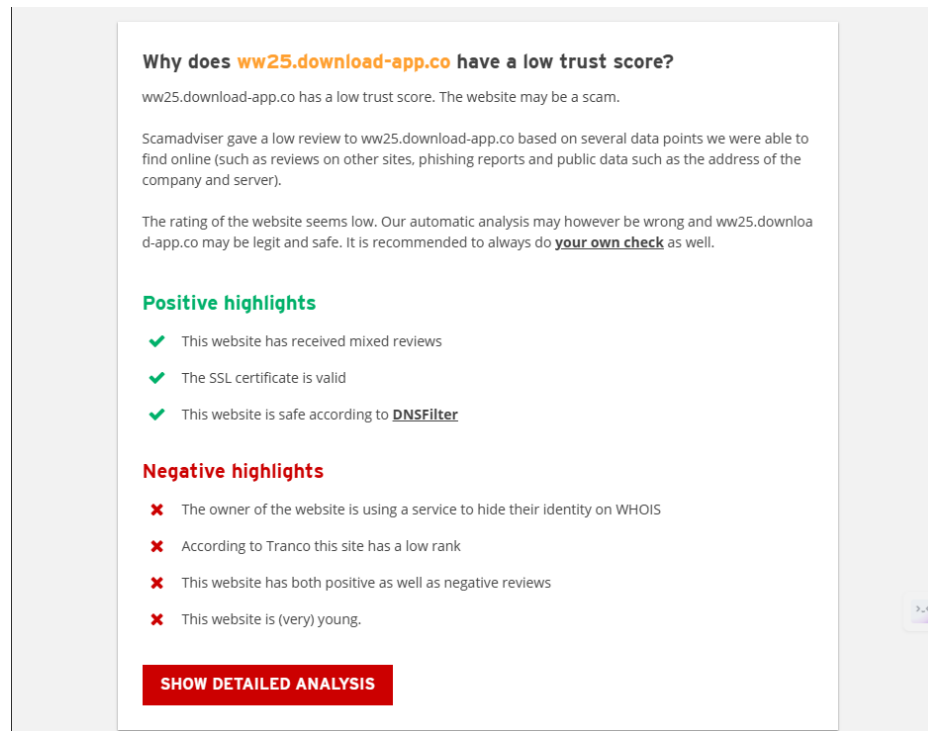


Figure 2.10 Positive and Negative Highlights of the Reviewed Website

In Figure 2.10 we can see the negative indicators mentioned in previous paragraph together with the positive indicators. These shows the good points of the website that suggest it may be trustworthy and the bad points of the websites that explains why its trustworthiness is reduced. In Figure 2 we can see the negative points explains that the owner of the website is using a service to hide their identity on WHOIS. WHOIS is a public database that stores information about domain names whenever someone registers it [27].

Table 2.1 Summary of Previous Work Done

Publisher	Types of Scams Targeted	Methods	Results
Derakhshan [16]	Telephone-based social engineering	Identify scam signatures with Natural Language Processing (NLP)	Achieve 90% and above in accurately identifying scams. No false positives after all the test
Roy et al. [17]	Phishing prompts (individual prompt detection, phishing collection detection, real-time phishing prompt subset detection, phishing email prompts)	Use a type of Natural Language Processing (NLP) model named RoBERTa to detect phishing prompts	Achieve a minimum 90% accuracy in identifying the four types of phishing prompts mentioned
Commercial Crime Investigation Department (CCID) Malaysia	All types of scams, general types of scams	Develop web application “Semakmule Portal” and mobile application “Check Scammers CCID” to allow citizens to check the validity of the information	Limited capability as the backend system relies on manual scam reports to update the database. Lacks of a system update to keep up with the later version of mobile system environment making it not available to newer version mobile smartphones

Truecaller	Spam information, calls, or messages	Primarily uses users' own contact lists and users' community reports to identify spam messages. Additionally, uses AI to analyse their database phone number search	<p>According to Trustpilot, the application performance is poor as many users reported that the application failed to block malicious phone numbers automatically [28]</p> <p>According to the Apple app store, it is rated 4.5/5 satisfaction with 245,800 user reviews which shows the application does not perfectly prevent spam phone numbers but majority of the time the application performed as expected [29]</p>
Whoscall	Malicious calls	Uses artificial intelligence and its own database to filter out spam	<p>Lack of information from Trustpilot</p> <p>According to the Apple app store, it has scored 4.8/5 satisfaction with 27,600 user reviews. The result shows that</p>

			the application has performed as expected [30]
ScamAdviser	Scam Websites	Uses attributes of the website to score the trustworthiness of the website	Fast and efficient way for the user to identify fraud website. According to Tladi from Make Use Of, his review states that the ScamAdviser's algorithm may not always be accurate. Some users did reported false positives [26].

To summarize all the previous work done, Table 2.1 provides information categorizing the content by publisher, types of scams targeted, methods utilized, and results.

2.1.4 Strengths and Weakness

Strengths

To use AI models to effectively detect scams, it will provide very great strengths in which if the AI model is strong and stable enough as tested by Derakhshan [16]. It was found that their AI model, particularly through the Natural Language Processing (NLP) techniques, detecting telephone-based social engineering attacks can yield high precision and recall performance, context-aware detection, continuous learning, and minimized false positives.

Truecaller and Whoscall has provided a very convenient way for the users to detect scams and spam calls. Their services are available on mobile smartphones allowing the users to quickly search for the source of the phone number and determine whether is the person on the other end legitimate. Both of them have their own large databases which all users would use to actively get the latest phone number database to automatically flag spam calls. Their application is also released internationally and does provide a solution to reduce the number of cases people fall victim to threat actors. Additionally, since the users are retrieving phone number databases from a centralized server, they would just need to update that master record and everyone's application would protect their users from the latest threats and scams.

ScamAdviser has also provided some of its key advantages. Firstly, it also provides an easy way to scan the input. Both web application and mobile application of the ScamAdviser would just need the user to input the URL of the website and it will conduct the scan. Then the detailed site information and the score of the website is displayed, which this is the second advantage. The second advantage is that the result of the scan shows the good and bad things found from the scanned website and the trustworthiness of the website. The final judgement of the reviewed website still lays on the hand of the user, the user could report or to leave a review on the result.

Weaknesses

To develop an effective scam detection AI model, the lack of data sets to train the related AI scam detecting models was one of the most common problems faced by the researchers. In [16] and [17] they had mentioned their AI was trained on limited amount of data which consist of scamming conversations between the scammer and the victims. According to [16], one of the reasons to the lack of data were due to the scam victims do not usually record their conversation or they are embarrassed to share their conversation which involves their financial loss.

On the other hand, the scam prevention solution released by the CCID Malaysia, Semakmule Portal web application and Check Scammers CCID mobile application , was not keep up to date. According to Figure 2.1.2, the mobile application Check Scammers CCID to verify the information was unable to support later versions of android which defeat the purpose to ease the process of verifying information by building a mobile version of the Semakmule portal. Meanwhile, the Semakmule portal which is built as a web application has limited capabilities to verify the information requested by the user. Limited information was available on how Semakmule's system operated but two facts can be confirmed is that the system was not utilizing any AI model to operate and the Semakmule's system works with a database [31]. Therefore, their database will be relying on manual scam reports to update their database, which will require large number of users willingly to contributing to its database to ensure the effectiveness of their verifying systems to counter scams.

As for Truecaller and Whoscall, both of the applications were also relying on the users report to contribute to their existing database to identify unknown caller phone numbers. The problem was if the unknown caller phone number was not previously reported, it will rely on the user's ability to identify whether it is a scam or not. If the person failed to identify the scam, he or she will becomes victims to the scammer and only after that has happen the unknown phone number would be flagged as a scammer phone number. Therefore, there exist inconsistency on preventing scams as the first person to encounter

CHAPTER 2

the scammer would have no prior information related to the unknown phone number despite having Truecaller or Whoscall installed on their smartphone.

Finally, for ScamAdviser, the application suffers from potential inaccuracies due to the algorithm limitations. The way that the algorithm works is by viewing the websites information such as popularity, social media activity, positive user reviews, performance, and security features. This would not be guaranteed accurate when one website is flagged as a low scored trustworthy website as there exist a chance that the website is just made by people with less experience in developing their website. This would have a very negative impact on those inexperienced developers and ruin their reputation before they could launch their product. Another problem that ScamAdviser is facing is that the mobile application of theirs are not available in every country, which means people in those countries are not accessible to their services via mobile and need to logon to their website to use their service, extra steps in verifying information sometimes could lead to some users feeling tedious.

CHAPTER 3 System Methodology/Approach

3.1 System Design Diagram

This section provides a high-level overview of the application's key features and structure. The goal is to offer a clear understanding of the system's functionality and how the various components interact within the overall design.

3.1.1 System Architecture Diagram

The following is the system architecture, the application developed consist of three major functions with each design with its own minor difference. Hence, three flowcharts have been shown below to briefly introduce the plan and process of each service offered by the solution.

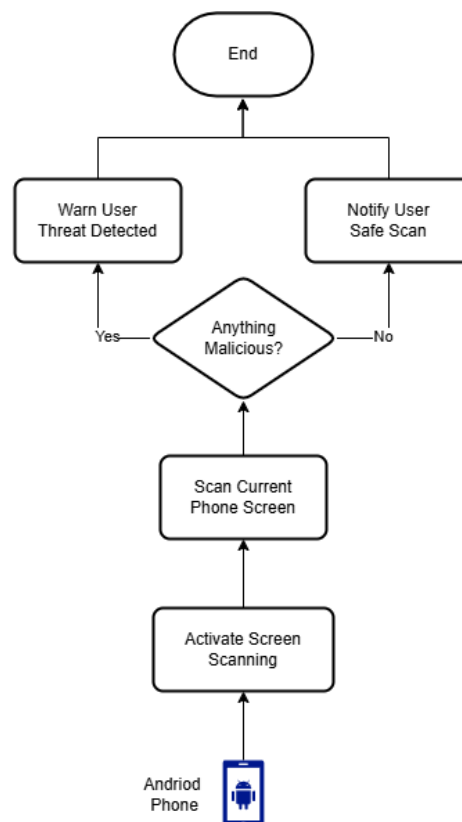


Figure 3.1 High Level Overview Screen Scanning Feature

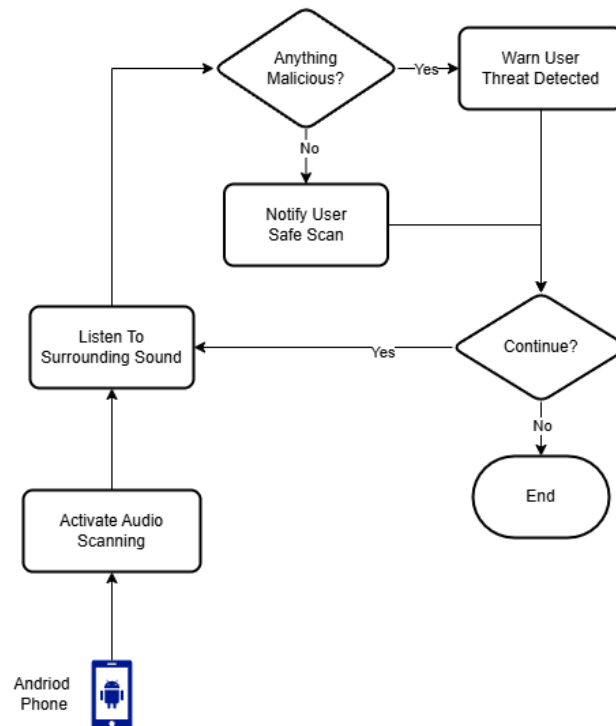


Figure 3.2 High Level Overview Audio Scanning Feature

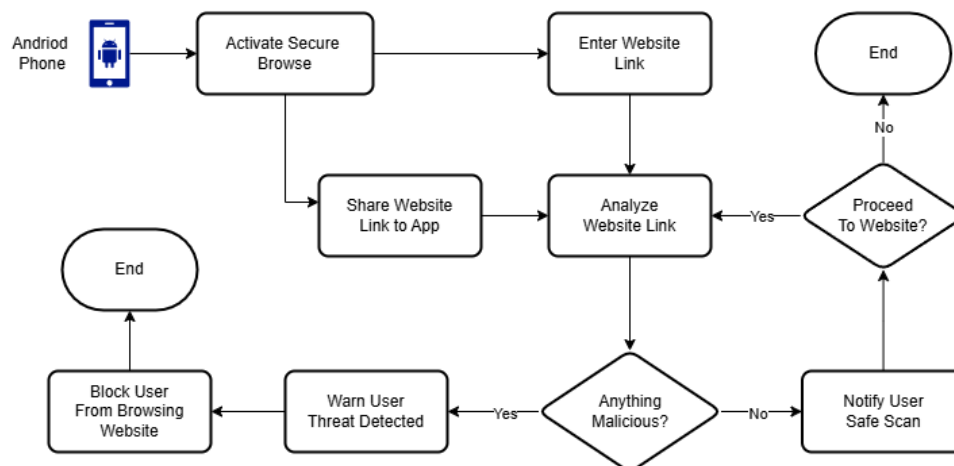


Figure 3.3 High Level Overview Secure Browse Feature

3.1.2 Use Case Diagram and Description

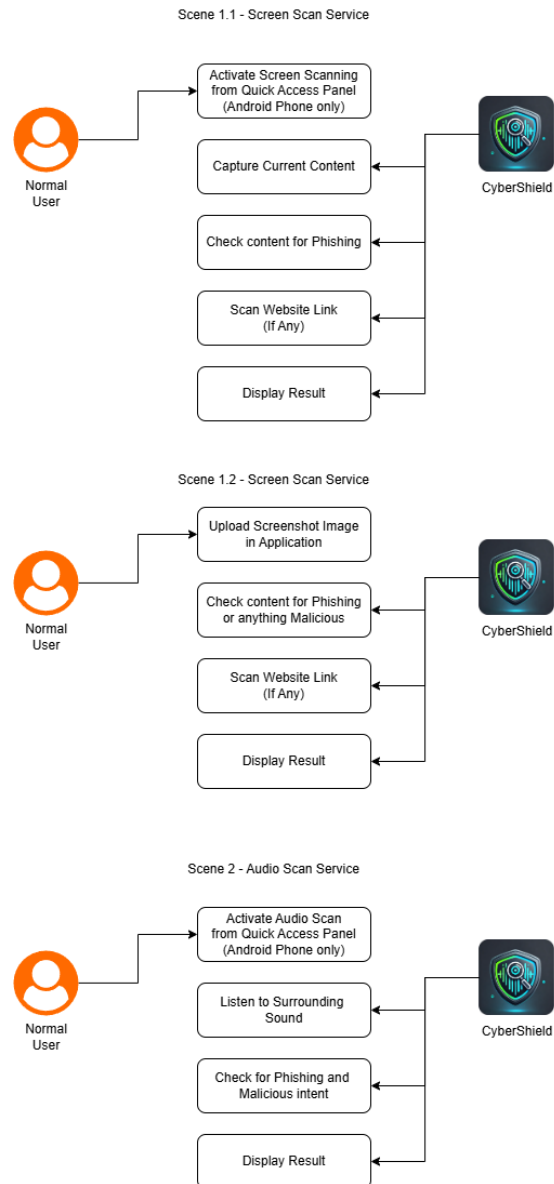


Figure 3.4 Use Case Diagram Part One

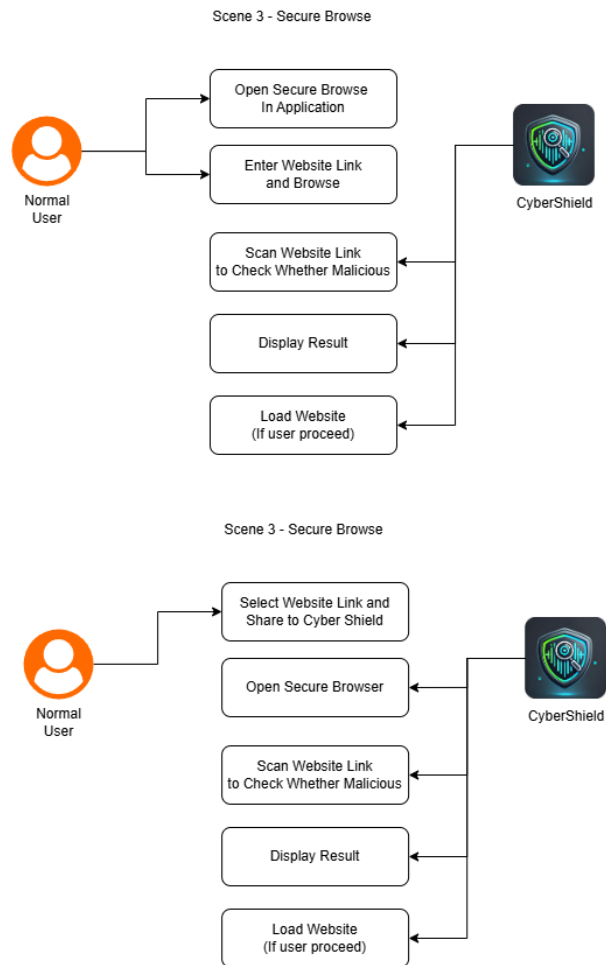






Figure 3.5 Use Case Diagram Part Two




3.2 System Requirements

3.2.1 Software

The project is mainly an application development and does not require any custom hardware to build the solution, thus, the project will utilize more heavily on the software services only. The following will be the list of software/programs that has been used to develop the solution.

Table 3.1 List of Software and Programs Used for Development

Tool	Role In the Project	Description
 Android Studio	Android Application Development Platform	One of the most common Android development platforms is Android studio. It is a robust and stable development platform; therefore, it has been chosen to be the primary place to develop the solution.
 Google AI Studio	Chatbot Gemini API Key Provider	Used to register and obtain the chatbot Gemini API Key.
 Gemini Gemini Chatbot	Decision Maker of the application logic	The chatbot Gemini model has been integrated into the application to serve as the decision maker of determining any malicious activity is spotted in the content.
 IPQS IP Quality Score	Provide IPQS API Enable Website Scanning Feature	The IPQS service provides an API that allows the application to have the capability to scan malicious website links and determining whether it is a threat to the user.

 <p>Google Machine Learning Kit Optical Character Recognition (OCR)</p>	<p>OCR service provider to the application</p>	<p>The Google Machine Learning Kit OCR allows the application to accurately convert image to text. It plays a crucial role in ensuring the final result of the scan to be as accurate as possible.</p>
 <p>Android Operating System</p>	<p>Primary Operating System to host the application</p>	<p>To ensure availability and accessibility while keeping the scope of the project feasible, the Android platform is chosen to be the candidate to host the solution.</p>
 <p>Java</p>	<p>Primary Programming Language Used for Application Development</p>	<p>The development of the mobile application will be coded in Java, a high-level programming language. Java mobile development has been a common choice among the mobile applications available in the market; hence, the Java mobile development community is large and sufficient to assist in the development of the source code of this project.</p>

CHAPTER 4

3.2.2 Hardware

There are two pieces of hardware that will be involved to assist in the development of this project which their details and specifications can be seen below:

Table 3.2 Specifications of Laptop

Description	Specifications
Model	MSI GF63 Thin 10UC
Processor	Intel(R) Core(TM) i5-10500H CPU @ 2.50GHz
Operating System	Edition Windows 10 Home Single Language Version 22H2 Installed on 17/11/2021 OS build 19045.4170 Experience Windows Feature Experience Pack 1000.19054.1000.0
Graphic	Nvidia GeForce RTX 3050 Laptop GPU 4GB DDR5
Memory	16GB DDR4 2667Mhz RAM
Storage	1.5Tb SATA SSD

Table 3.3 Specifications of Smartphone

Description	Specifications
Model	Xiaomi Poco X3 NFC
Brand	Xiaomi
Processor	Qualcomm SM7150-AC Snapdragon 732G (8 nm)
Operating System	Android 14, MIUI 14
Graphic	Adreno 618
Memory	8 GB RAM
Storage	128 GB

3.3 System Design

The development of the system would be separated into two sections: the front end and the back end. The backend purely focuses on data processing, such as converting screenshots taken to text. On the other hand, the front end allows the user to interact with the mobile application easily; it displays the available functions that the mobile application provides.

In this project, it is more focused towards the methodology of how to protect the user from being scammed. The backend is the critical component that is developed to counter the disinformation causing the majority of the scamming cases. The backend system design could be simplified into one main system that manages crucial services and interconnects them. It forms the entire process, starting from taking the input to displaying the result to the user. The crucial services that the application uses are Google Gemini LLM API, IP Quality Score API, Google Machine Learning Kit OCR, and some other services that are built-in in Android to assist in retrieving data, such as the screenshot API from Android.

In order to ease the management of the services, the services are coded into modules that simplify the main system managing them. Referring to Figure 3.1.1, where the figure shows the flowchart related to the process of screen scan functionality, all the processes work independently, and the processes would just take the data from the previous process, then continue the streamline to process and throw it to the next component to process. Ultimately, the final result will be shown to the user. Additionally, Figure 3.1.2 shows how the audio listen functionality works. The difference is insignificant; the input has changed to audio input, and there is no URL checking for this audio listen functionality flow.

This section should demonstrate how the screenshot is converted to text, how the text is thrown to Gemini for processing, and how Gemini retrieves the conversation transcript from the user.

CHAPTER 4

3.4 Project Milestones

3.4.1 Project 1 Milestones

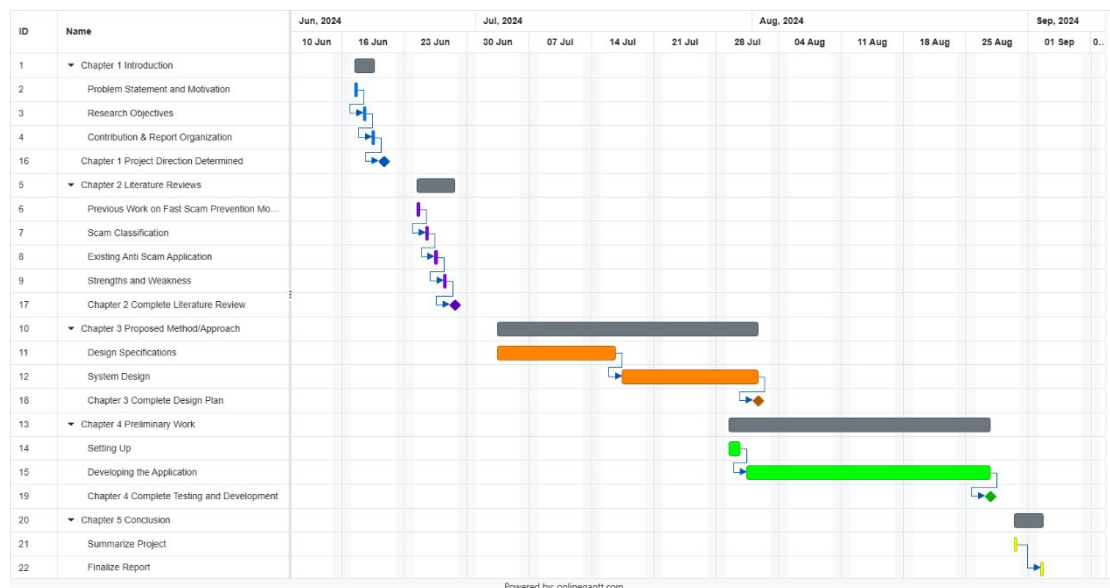


Figure 3.6 Project I Milestones

Planning for the Project 1 Period

The plan for the project 1 period (4 months) is to fully develop the application to be equipped with all the functionality of taking the audio (surrounding sound) or visual (the smartphone's current screen) to analyze the content. Then it would produce the output whether the content is safe or dangerous. The output is a pop up window to guarantee that the user does not overlook at the result of the analysis and effectively warns the user if any dangerous content was spotted. The application is built minimize the human interaction required to operate the services that it provides to ensure ease of use. The planning for project 1 period has successfully completed all of the mentioned.

3.4.2 Project 2 Milestones

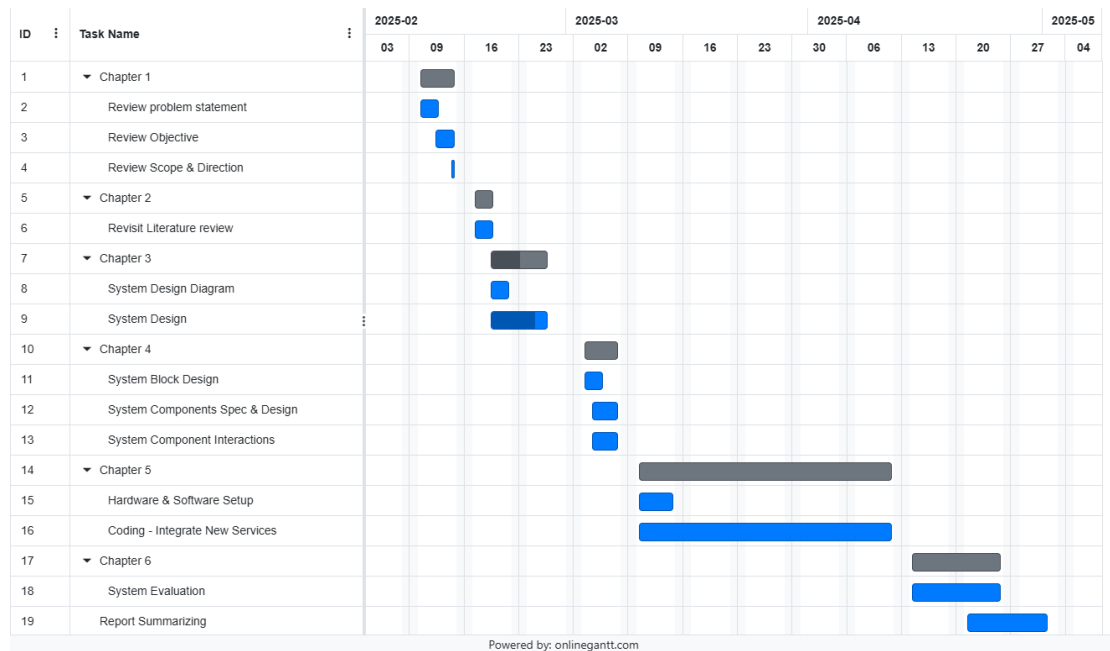


Figure 3.7 Project II Milestones

Within project II, the milestones can be seen in Figure 3.3. The main objective of project II period is to complete the development of the entire application including a more user friendly user interface. Additionally, project II will be working on enhancing the function of the existing ones such as screen scan and audio scanning to make it more robust and resilient.

3.5 Estimated Cost

The estimated cost of the project at this current stage is zero in terms of monetary. All the services that have been utilized do not cost anything as they are under the free tier of their service. Indirect costs such as utility cost and time cost are not accounted for, the period of the entire development is estimated to be 1 year.

However, if the application were to scale accordingly to account for large amount of user the cost to maintain the service will not be significant as the cost will only accumulate for Gemini API and IPQS API only.

3.6 Concluding Remark

This chapter we have introduced the high level overview of the services that will be provided by the solution. Aside from that, the system requirements are also briefly mentioned to let the readers have a rough estimation of the hardware and software required for the solution development if they wanted to replicate the entire project. The project milestones are available in section 3.4 where the planning of the project development can be seen.

CHAPTER 4 : System Design

Within this chapter, it will include and provide more explanations for the overall system design with further technical details.

4.1 System Block Diagram

The following are three detail diagrams that explain the technical of the three major core services that are provided by the application.

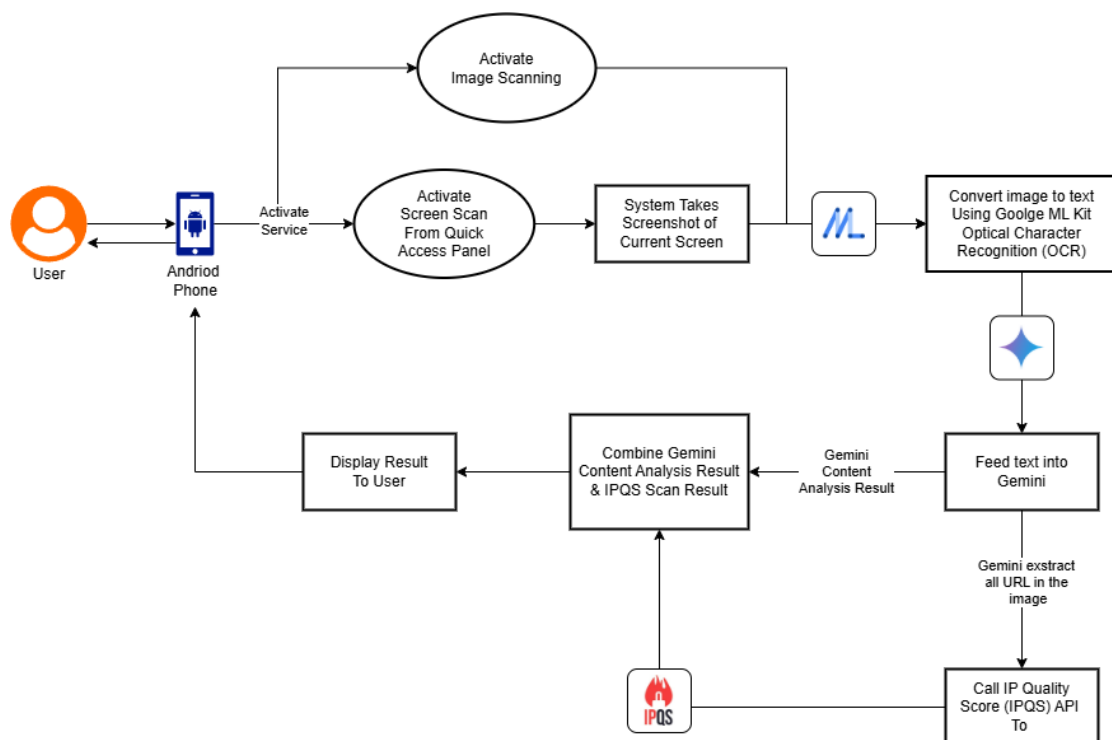


Figure 4.1 System Design of Screen Scanning Service

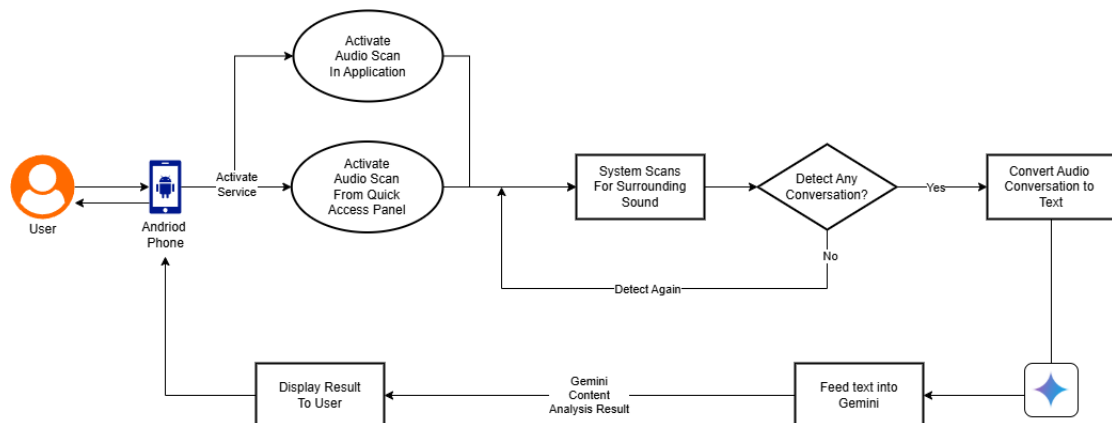


Figure 4.2 System Design of Audio Scanning Service

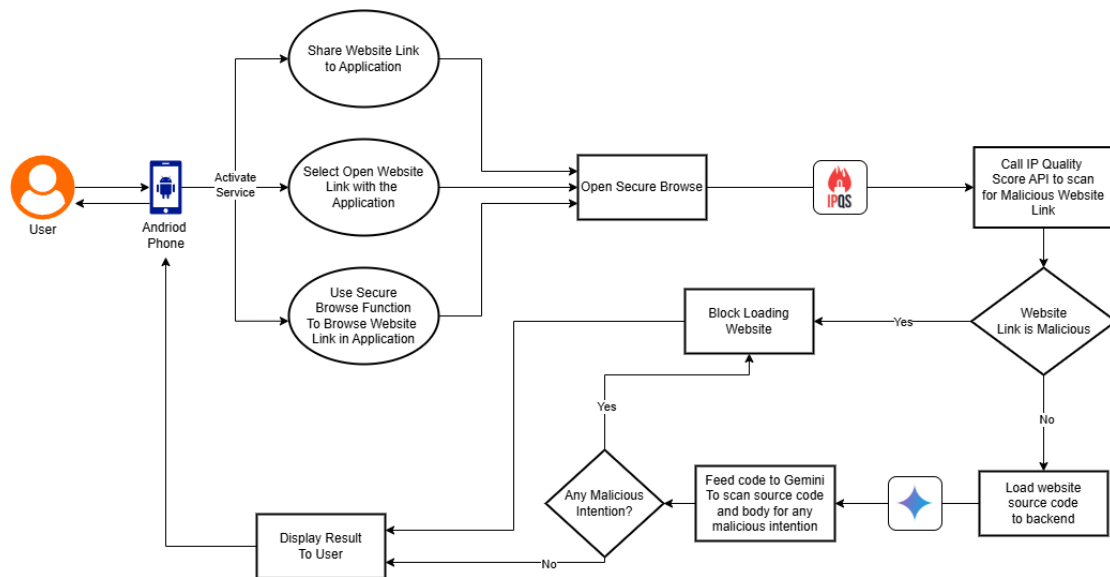


Figure 4.3 System Design of Secure Browse Service

The following is a quick introduction to the components and services utilized to build the three major functions of the application.

User

- The user plays a crucial role in assisting the application to detect any phishing or malicious intent that may be a threat to the user. The development of the application has taken into consideration of user's privacy; hence, the activation of the application services is active instead of passive. This means that the user will need to manually activate the application services with one or two methods provided by the application. For example, activating the screen scanning service via the quick access panel of the Android Smartphone device.

Android Smartphone

- To utilize the application, the user must possess an Android Smartphone with Android 14 or above. Since the application is built for Android and tested with Android OS version 14. The user should meet the requirements to have the best possible experience of the application.

CHAPTER 4

IP Quality Score API

- The IP Quality Score or known as IPQS, has a service that allows developers to integrate a malicious website link scanner to their solution by Application Programming Interface (API). This has migrated the need for the application to build and maintain a website link scanner.

Gemini API

- The Gemini Chatbot API is a chatbot that is developed by Google. It plays a crucial role in the application as it is the decision maker of determining whether the content scanned is malicious or safe.

Google Machine Learning Kit Optical Character Recognition (OCR)

- One of Google's solutions to Optical Character Recognition (OCR) is its Google Machine Learning Kit OCR. Compare to Tesseract which was also made by Google, Google Machine Learning Kit OCR is more accurate for this project's use case.

4.2 System Components Specifications and Design

In this section, we will review the three major technologies that has been utilized in the project to form the services that the application offers. This includes the image to text conversion tool Google Machine Learning Kit OCR, Gemini chatbot for decision making, and IP Quality Score for scanning malicious website links.

4.2.1 Converting screenshot to text

The idea of converting the screenshot to text instead of making the user uploading their own screenshot is intended to ease the process of screen scanning and validating the information on the current smartphone screen. However, this process proposes some problems. Firstly, the process of converting image to text could be unnecessary as the current Gemini models are able to interpret images and provide services such as describing or answering questions related to the image, summarize the content, or extrapolate from the content [32]. Secondly, the resolution of the screenshot is limited to the actual hardware screen from the smartphone device, which means the input image resolution may be affected and limited to what the user's smartphone screen could provide. Thirdly, running an offline local Optical Character Recognition (OCR) model on the smartphone could introduce more load the smartphone hardware, which may delay the response time of the application.

The experimentation of the prototype was built with an offline OCR model to test the performance of the model. The initial model that was chosen to perform the task was the Tesseract OCR model [33].

Optical Character Recognition (OCR)

Before viewing the Tesseract architecture, it is important to get an overview of the OCR architecture. The Optical Character Recognition (OCR) is the process of converting an image to text. The OCR engine would generally work in the following sequence [34]:

1. Image acquisition

The scanner will read the image and convert it into binary data. The OCR engine will analyse the scanned image and classify the light areas as background and dark areas as text.

2. Preprocessing

The OCR engine will clean the image and remove errors to prepare it for scanning. This process involves slightly instilling the scanned image to fix alignment problems, removing any digital spots or smoothing the edges of the text images, cleaning boxes and lines in the images, and more.

3. Text Recognition

Two primary types of OCR algorithms exist that an OCR software uses for text recognition. The first is pattern matching, and the second is feature extraction.

3a. Pattern Matching

This method involves separating a character image and compares with a stored character image. Pattern recognition works only if the stored character image has a font and scale similar to the input character image. This method works better with images that contain a common text font.

3b. Feature Extraction

This method will break down the character image into features such as lines, closed loops, line direction, and line intersections. It will then use these features to find the best match or nearest neighbour among its stored character images.

4. Postprocessing

After the analysis, the system will convert the extracted text into a computerized file.

Tesseract OCR model architecture

The simplified version of the Tesseract working principle is as follows [35]:

1. Input Preparation

The model expects a binary image (black and white) as its input, usually with defined polygonal text regions. This will remove the need to separate another page for layout analysis. Polygonal text regions refer to the areas in the image where the text is located. The marked area is typically shaped like polygons hence it is known as polygonal text regions.

2. Connected Component Analysis

The model will start by identifying the connected components in the image, focusing on the outlines of the characters and other graphical elements. The outlines refers to the specific lines or curves of the shape of the character.

3. Blob Organization

After successfully identifying the outlines, the model groups them into "blobs" and organizes them into text lines. Then, the model analyzes text lines based on the fixed or proportional character spacing.

4. Recognition Process

The model uses a two-pass recognition process. The first pass recognizes each word sequentially; satisfactory recognitions are used to train an adaptive classifier. The second pass involves words that failed to be recognized in the first pass and are revisited for improved recognition.

5. Adaptive Classifier

The model uses a static classifier to process general character recognition but will utilize an adaptive classifier that improves accuracy by learning from the recognition results of the previously identified words.

6. Linguistic Analysis

The recognition engine applies minimal linguistic analysis to enhance the recognition process. It will consider the most probable words based on the context and frequency indicators, contributing to refining the recognition results. The linguistic analysis consists of having the model using linguistic rules and statistical models to improve the accuracy of text recognition.

7. Final Adjustments

The final stage will include resolving fuzzy spaces and verifying potential alternatives to recognize characters correctly. This stage also includes adjustments that account for small-cap text and other nuances in font styles.

The Tesseract model is a common and open source model that developer use to recognize text. However, upon testing, the Tesseract model performance on our smartphone was not up to expectation and constantly have trouble on properly recognizing characters and symbols that are similar to each other. For example, the letter “L” in its lower case form is constantly being recognized as a pipe symbol. It was found that this was an issue that was faced by some other developers, one of them suggested that the model was recognizing the letter ‘L’ lower case as pipe in 100 percent of the time [36]. One solution is to always convert the pipe symbol to the letter ‘L’ lower case, but it still does not address the accuracy problem of the model. The development of the application has attempted to fix the issue by changing the image resolution to 300dpi which was recommended when using this model [12]. Unfortunately, the problem persisted which has led to finding alternative solution or OCR models that will provide more solid result.

On the other hand, despite the performance of Tesseract model was not up to expectation, throughout the process, it can be concluded it was better to do the image to text conversion process locally by installing local OCR instead of uploading the image to the LLM model, this is because all LLM available on the Internet will charge the user more if they want to process image compared to processing only the text. Then the screenshot resolution can be adjusted to 300dpi which was already done to test the Tesseract model. Finally, the load on the smartphone is unnoticeable.

The next candidate that is tested to become the OCR model in the application is the Google Machine Learning Kit OCR [37]. This model is a very convincing OCR model as the tested result on the prototype solves the accuracy problem suffered by the Tesseract model.

Google Machine Learning Kit Text Recognition version 2

The model is an on-device machine learning expertise to Android and IOS apps. The text recognition model is based on Google's Machine Learning kit where the machine learning will utilize vision and natural language APIs to solve the proposed problems. One of the problems that it is capable of solving is recognizing text from images [38]. The text recognition model is able to recognize multiple text such as Chinese, Devanagari, Japanese, Korean, and Latin. It will analyze the structure of the text and detect symbols, elements, lines, and paragraphs. It is also capable of real-time recognizing text and the language of text [39].

The recognition model will recognize text by segmenting the text into blocks, lines, elements, and symbols. The words can be defined as the following:

Block : A contiguous set of text lines, such as paragraphs or columns

Line : A contiguous set of words on the same axis

Element : A contiguous set of words on the same axis in most Latin languages

Symbol : A single alphanumerical character on the same axis

CHAPTER 4

One great example to visualize how the Google recognition model recognizes the text is as following.

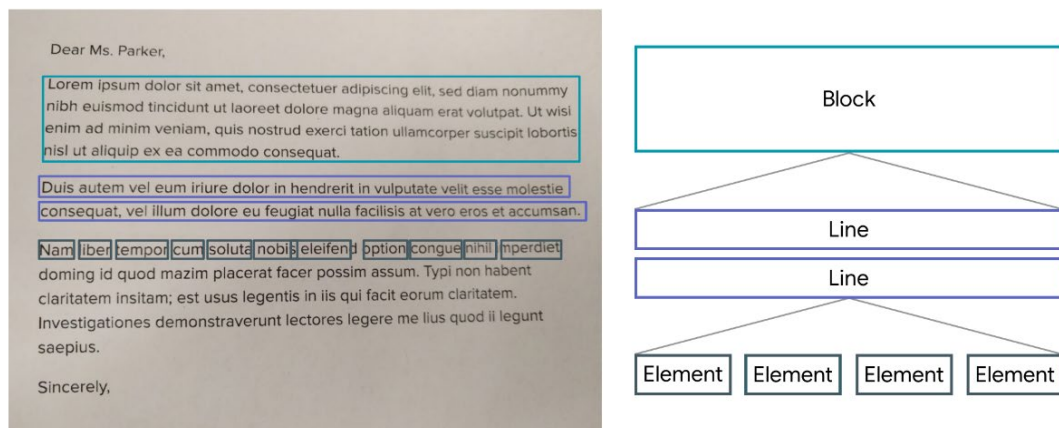


Figure 4.4 Google Recognition Model Recognizing Text [39]

The Figure 4.4 is obtained from Google's own text recognition model guide. The first highlight in cyan is a block of text. Then the blue highlighted box is a line of text. Then the dark blue boxes are words. All the detected blocks, lines, elements, and symbols will be analyzed by the model and will use the API to return the bounding boxes, corner points, rotation information, confidence score, recognized languages and recognized text.



Figure 4.5 Google Guide Example OCR Image [39]

The Figure 4.5 includes the example image input to the Google recognition model and the result shows

Recognized Text	
Text	Wege der parlamentarischen Demokratie
Blocks	(1 block)
Block 0	
Text	Wege der parlamentarischen Demokratie
Frame	(296, 665 - 796, 882)
Corner Points	(296, 719), (778, 665), (796, 828), (314, 882)
Recognized Language Code	de
Lines	(3 lines)

Figure 4.6 Google Guide Example OCR Result part 1/3 [39]

CHAPTER 4

Line 0	
Text	Wege der
Frame	(434, 678 - 670, 749)
Corner Points	(434, 705), (665, 678), (670, 722), (439, 749)
Recognized Language Code	de
Confidence Score	0.8766741
Rotation Degree	-6.6116457
Elements	(2 elements)
Element 0	
Text	Wege
Frame	(434, 689 - 575, 749)
Corner Points	(434, 705), (570, 689), (575, 733), (439, 749)
Recognized Language Code	de
Confidence Score	0.8964844
Rotation Degree	-6.6116457
Elements	(4 elements)

Figure 4.7 Google Guide Example OCR Result part 2/3 [39]

Symbol 0	
Text	W
Frame	(434, 698 - 500, 749)
Corner Points	(434, 706), (495, 698), (500, 741), (439, 749)
Confidence Score	0.87109375
Rotation Degree	-6.611646

Figure 4.8 Google Guide Example OCR Result part 3/3 [39]

In Figure 4.6, Figure 4.7, and Figure 4.8 we can see that it follows the architecture explained in Figure 4.4. Overall, Google's Recognition Model proposes a more robust and stronger OCR model compared to Tesseract OCR model.

After implementing Google's OCR model, the following is one test case to show the flow of the image to text conversion process:

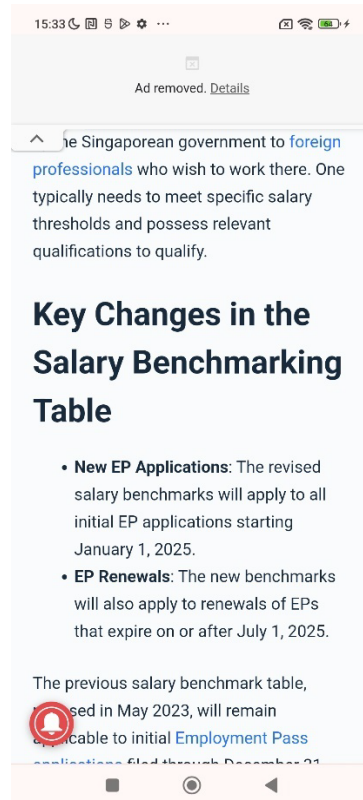


Figure 4.9 The Current Smartphone Screen

CHAPTER 4

1. Capture the screenshot

```
File location      com.example.fyp_ver_2      I      /storage/emulated/0/Android/data
ScreenCaptureService com.example.fyp_ver_2      E      captured image: 0
```

Figure 4.10 System Log Capture Screenshot Success

2. Google Machine Learning Kit OCR running

```
googleOCR          com.example.fyp_ver_2      E      googleOCR result : 15:34 G 0
Ad removed. Details
he Singaporean government to foreign
professionals who wish to work there. One
typically needs to meet specific salary
thresholds and possess relevant
qualifications to qualify.
Table
C64 4
Key Changes in the
Salary Benchmarking
• New EP Applications: The revised
salary benchmarks will apply to all
initial EP applications starting
January 1, 2025.
• EP Renewals: The new benchmarks
will also apply to renewals of EPs
that expire on or after July 1, 2025.
The previous salary benchmark table,
sed in May 2023, will remain
cable to initial Employment Pass
StartMainScreenshot com.example.fyp_ver_2      I      Google OCR ML Kit completed.
StartMainScreenshot com.example.fyp_ver_2      E      Finishing activity.
```

Figure 4.11 Google Machine Learning OCR Result

3. Load on smartphone

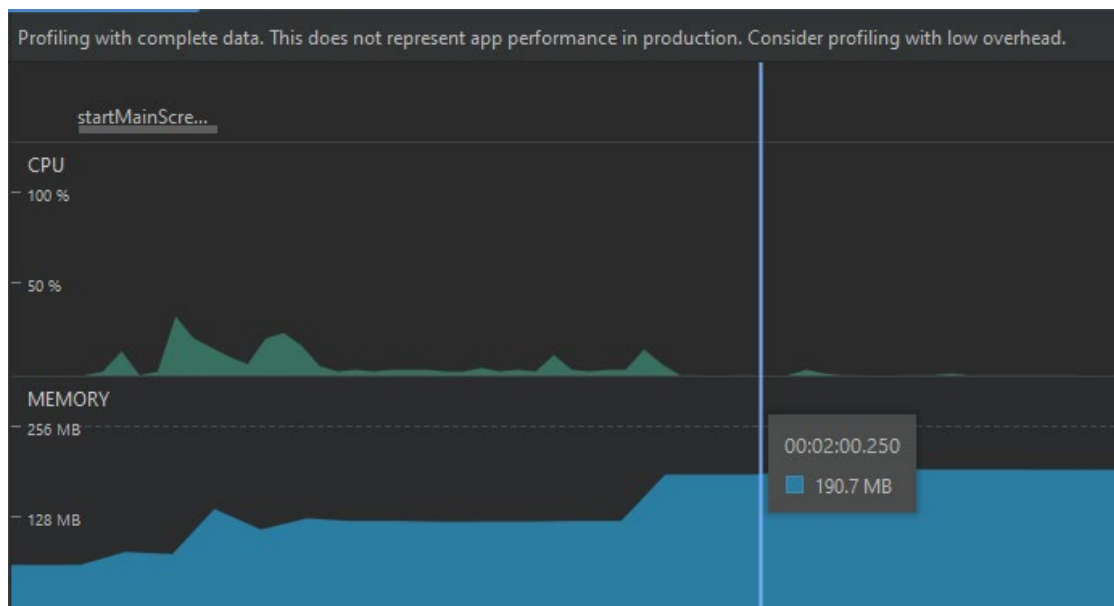


Figure 4.12 CPU and Memory Usage Text Conversion Process

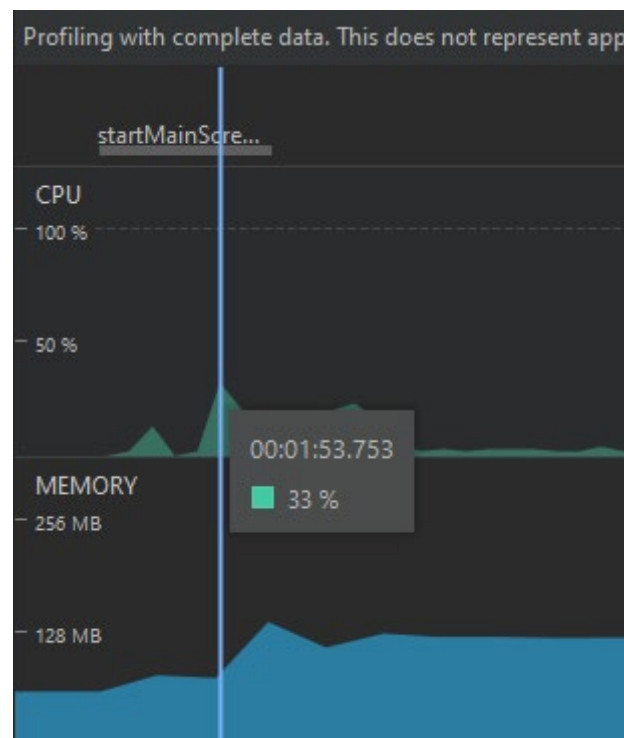


Figure 4.13 Detail CPU and Memory Usage Text Conversion Process

After testing, it can be seen that Google OCR has performed up to the expectation and is sufficient for our use case. The project is also taking into account that as long as the OCR model that is implemented can convert the image to text to a level that the LLM is able to automatically correct the wrongly recognized word then it would be sufficient for this project's use case.

The performance hit on the smartphone that it was tested on, which is the Xiaomi Poco X3 NFC with the CPU hardware of Qualcomm SM7150-AC Snapdragon 732G, is shown in Figure 3.2.9 and Figure 3.2.10. Both of the figure shows that the CPU usage is able to be tolerated as the peak usage is only at 33% and the memory (RAM) usage is roughly 128Mb only. Hence, Google Machine Learning Kit OCR model is a very good candidate for our use case and will be the chosen model to perform this task.

4.2.2 Google Chatbot Gemini

Gemini 2.0 is Google's latest artificial intelligence innovation; it is a leap forward from earlier unimodal models [40]. Being a multimodal artificial intelligence system, the chatbot can process and understand various forms of input. These include text, images, audio, video, and code.

The multimodal ability allows it to perform more varied tasks with more accuracy and flexibility than those previous traditional models limited to text data. Gemini 2.0 is programmed to handle hard reasoning, inference, and problem-solving in a wide range of domains with high efficiency and scalability.

The chatbot is based on advanced deep learning techniques employing the Transformer framework as its foundation. Its operation modes can be summarized as below:

Transformer Architecture: The Transformer network at the heart of Gemini 2.0 identifies relationships and dependencies within each modality as well as across modalities.

Self-Attention Mechanism: With this mechanism, Gemini 2.0 is capable of focusing on the most pertinent parts of the input and, for example, linking descriptive text to the corresponding objects in an image.

Cross-Modal Attention: By synthesizing information from different kinds of data, Gemini 2.0 can better comprehend how, for example, text relates to an image, or sound is coordinated with a video.

Contextual Understanding: Through a series of Transformer layers that input goes through, Gemini 2.0 accumulates a rich, contextual sense that is required for reasoning, decision-making, and generating coherent outputs.

Output Generation: After processing, the model can generate a range of outputs from text and images to code by mapping its internal representations into human-readable form.

Transformer Architecture

While there is no official documentation regarding the actual Gemini 2.0 transformer architecture, we will take an overview on the base transformer architecture. The core of the transformer model is the self-attention mechanism, this mechanism allows the model to weigh the importance of each element in the input sequence. Thus, those transformer models are able to show a more context-aware understanding towards the data compared to their predecessors [41].

In Natural Language Processing (NLP), Large Language Models like GPT and BERT (Gemini predecessor) utilize the self-attention mechanism to capture long-range dependencies and context. As a result, their benchmark shows superior performance and one of it, shown in our project, text generation.

Transformer is a deep neural network that perform analysis on the sequence of the data, keep tracks of the context and finally generate a corresponding new data. These models are trained on a large amount of unlabelled data in a self-supervised manner. Self-supervised learning is a deep learning approach where a model is pre-trained using unlabelled data by creating artificial labels derived from the data itself. This process helps the model learn useful representations without requiring manually labelled data [42]. The following figure is the main components of the transformer model from the initial transformer research team in 2017 by 8 researchers from Google.

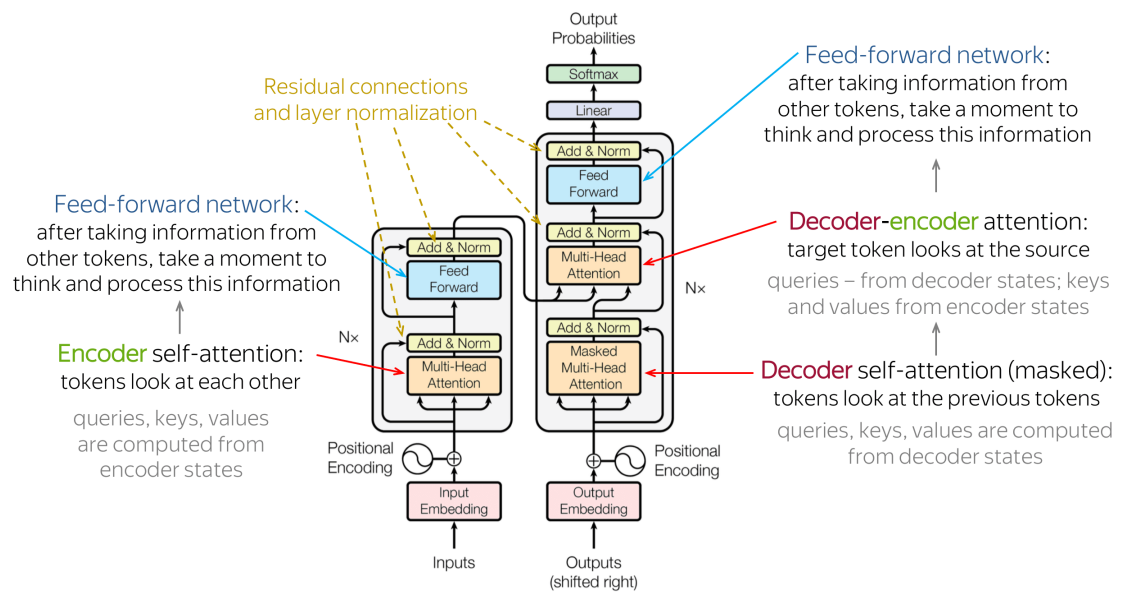


Figure 4.14 Transformer Model from Original Paper [43] with Additional Comments [41]

The transformer key elements:

Encoder-Decoder : The transformer model has two main parts which are the encoder and decoder. Both of the elements are made up of repeating layers.

Encoder takes the input text and generates a numerical representation of it. Each encoder layer will process the input and pass to the next encoder. Decoders generate the output text step-by-step. Each of them receives information from the final encoder and the previous decoder layer.

Multi-head Attention Mechanism : For both encoder and decoder will consist of this self-attention mechanism to get an estimation of the importance of different words in the sentence. This will allow the model to understand the relationship between each word in the input no matter how far apart those words are located [41]. Thus, the model is able to pay attention to multiple parts of the sentence simultaneously.

Feed Forward Neural Networks : For both encoder and decoder will also consist of this mechanism. This mechanism will allow the model to add complexity to the word representation to help it learn more complex patterns.

Cross-Attention: This mechanism allows the decoder focus on the relevant parts of the original input text while it's generating the output.

Positional Encodings: Positional encoding add information about the position of each word in the sequence, so the model understands the order of words.

Residual Connections: Residual connections help the model remember the original input and avoid forgetting information during training.

Layer Normalization: This technique makes the training process more stable and helps the model learn better.

Softmax Layer: This final layer converts the model's output into probabilities, allowing it to predict the next word in a sequence.

By combining the key elements, we are able to form a transformer model workflow.

Step 1 Tokenization and Embedding : Input text is segmented into a tokens and every token mapped onto a vector of numbers (embedding) reflecting its semantic representation. Position information is injected in the embedding in order to pick up on word order.

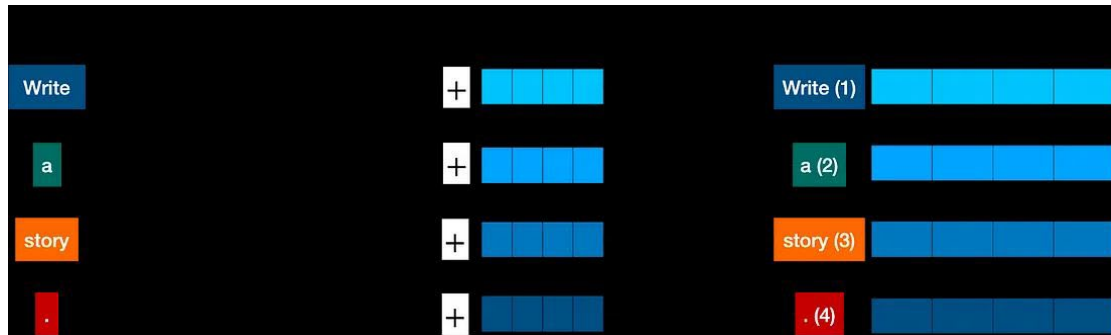


Figure 4.15 Tokenizing Words [44]

Step 2 Encoding: The tokens inserted are processed by the encoder. Self-attention in the encoder allows each word to look at all the other words in the input, acquiring the context knowledge. Feed-forward networks also process these contextualized representations.

Step 3 Decoding : Decoder generates the output sequence depending on the encoded information step by step. Masked self-attention prevents the decoder from "seeing" words that appear later. Cross-attention allows the decoder to look at corresponding parts of the encoded input. Feed-forward networks are utilized here too.

Step 4 Output Generation : The final output of the decoder is passed through a linear layer and a softmax layer. It converts the vectors into probabilities for each word in the vocabulary and chooses the word with the highest probability as the next word in the output.

In short, the flow:

1. Inputs get tokenized
2. Tokens convert to numbers (embeddings)
3. Model keeps track of the order the token appears (positional encoding)
4. Token is represented as a vector (a list of numbers), then fed back into the model
5. Encoder reviews the tokens, uses self-attention to understand how each words are related to each other. Then processes them through a neural network to get a deeper understanding of the text (contextual representations)
6. Decoder retrieves the understanding, and focus on the crucial parts only (cross-attention), then generate the output, one word at a time
7. Finally, the model uses a simple layer to turn the output into a set of possibilities (logits), and a softmax layer will pick the most likely next word or symbol.

4.2.3 IP Quality Score

The IP Quality Score (IPQS) which is the module that will be providing URL scanning service via API. The IP Quality Score (IPQS) URL Scanner API for Malicious URLs will scan the links and send to the API real-time to flag suspicious URLs [45]. The service will flag poor reputation domains, suspicious links, and phishing URLs with a real-time API that can be implemented in other application or projects. The service can classify over 70 website categories for easier analysis of unknown sites. A few of the more important categories that it can detect for our use case are phishing, malware, command and control, and parked domain websites. The Command and Control (C2) URLs allow an attacker to communicate with botnet servers. The parked domains refer to domains that have a similar spelling to brand names with minimal typos; therefore, it will confuse consumers and lure them to their website.

4.3 Sample System Components Interactions

The following is a test run for running the Google OCR, Gemini analysis and the IPQS URL scanner to demonstrate the three major system components collaborating to form the screen scanning service.

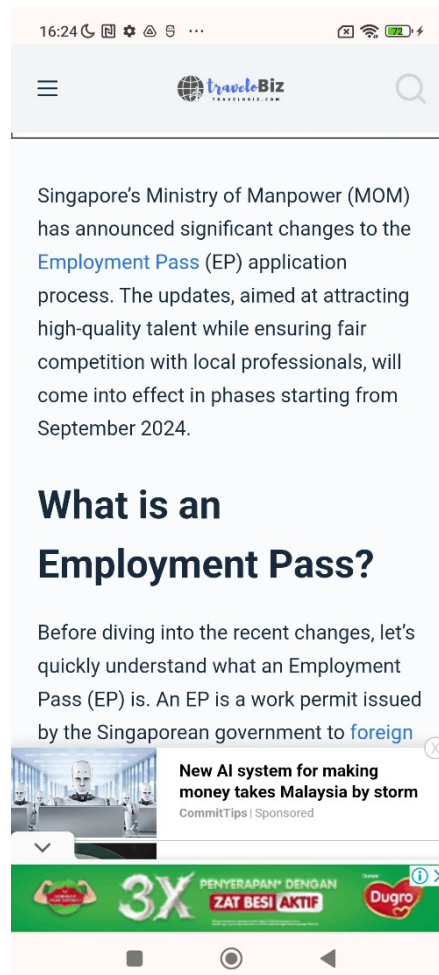


Figure 4.16 Sample Picture



Figure 4.17 Gemini Result

Gemini Result : Output: Safe

Description:

The text describes changes to the Employment Pass application process in Singapore. While this is a significant update, it is a legitimate announcement from a government agency. The information is readily available on the Ministry of Manpower's official website.

Another attempt is to show the URL scanning process.

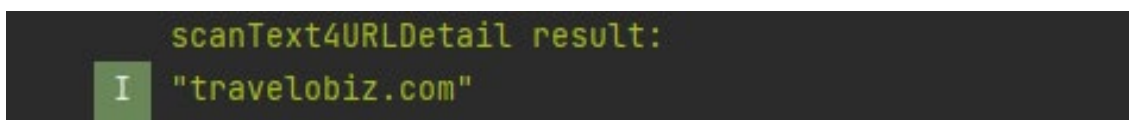


Figure 4.18 URL Found

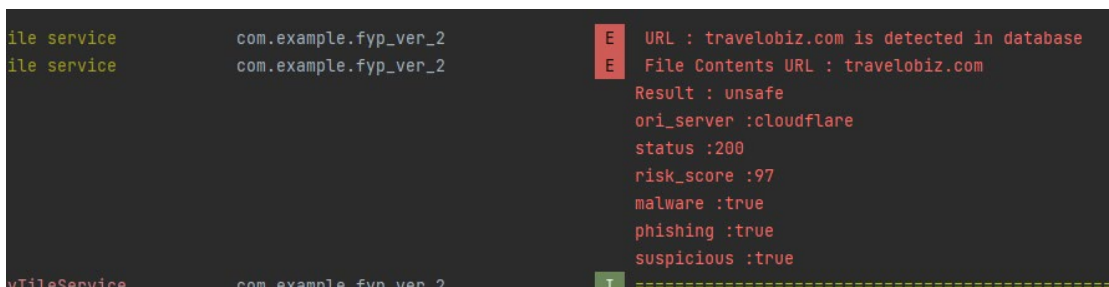


Figure 4.19 IPQS Scan Result

In short, the entire process of pushing the data to Gemini and letting it to judge the content of the data is very feasible as it is tested using the prototype application.

CHAPTER 5 System Implementation

5.1 Hardware Setup and Configuration

This project does not require any special hardware setup as it is more software oriented development. Therefore, the hardware setup will only consist of two components listed below:

Android Smartphone



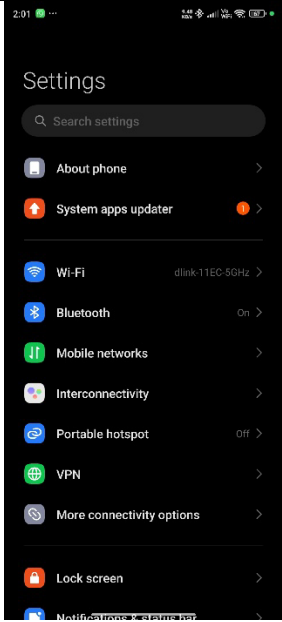
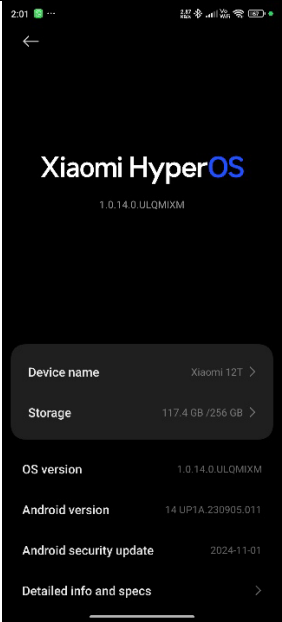

Figure 5.1 Image of Xiaomi Poco X3 NFC

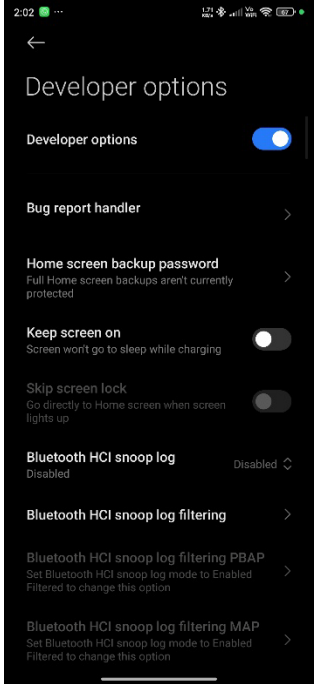
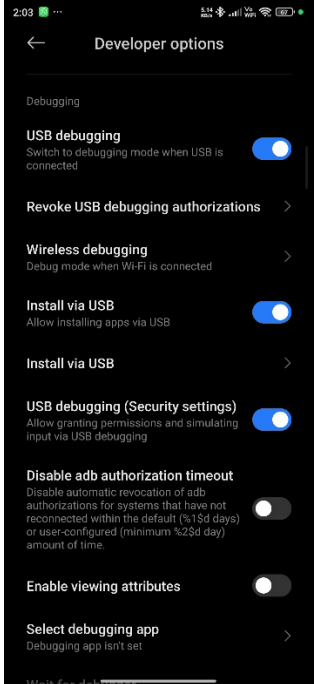
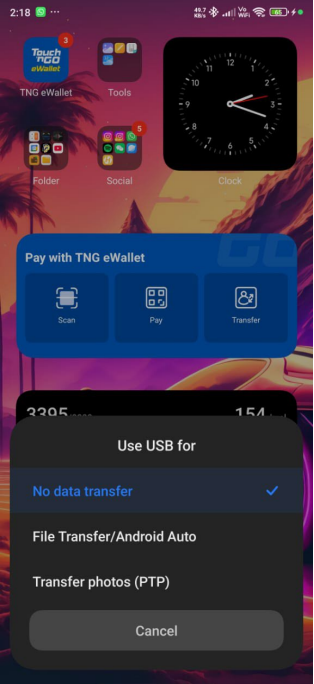
The use of the Android smartphone is to host and debug the application developed.

The development is built for Android platform and tested on Android version 14. Hence, the requirements replicating the development of the project is set to Android 14 smartphone. It should be noted that older variants of the Android version may work to host the application; however, since it is not tested for any older variant of Android system, there is no guarantee the system will work as expected.

Aside from that, the smartphone will require one configuration to allow debugging mode later on. For this project, the Android Smartphone used for deploying the application is the Xiaomi Poco X3 NFC with Android 14. The configuration can be seen at the following table:

Table 5.1 Phone Configuration

Step 1: Go to the smartphone settings	Step 2: Go to the About phone	Step 3: Click the OS version label until you are developer (May be different for each phone brand)
		
Figure 5.2 Phone Configuration Part I	Figure 5.3 Phone Configuration Part II	Figure 5.4 Phone Configuration Part III

Step 4: You should have a developer option page available in your setting, navigate to it.	Step 5: Ensure the “USB debugging”, “Install via USB”, “USB debugging” are enabled	Step 6: When plug in the phone via USB to the computer, ensure the phone configure the USB for File
 <p>Figure 5.5 Phone Configuration Part IV</p>	 <p>Figure 5.6 Phone Configuration Part V</p>	 <p>Figure 5.7 Phone Configuration Part VI</p>

Computer



Figure 5.8 MSI GF63 Thin 10UC

The computer can be any computing device with Windows operating system 64 bit. This includes laptops or desktops as long as it is able to support all the programs that are introduced in the software setup later on. For this project, the computer that is utilized to build the project is a MSI GF63 Thin 10UC with Windows 11 home edition 64 bit operating system. No configuration is needed on the laptop out of the box.

5.2 Software Setup and Configuration

5.2.1 Android studio

Android Studio is a software that requires the user to download in order to use its service. Due to multiple versions of the Android studio is available and updates for the platform are frequent, the project's Android studio version is set at Ladybug Feature Drop 2024.2.2 Patch 1, released on February 13, 2025. Users may download the Windows 64-bit variant and proceed with the standard installation.

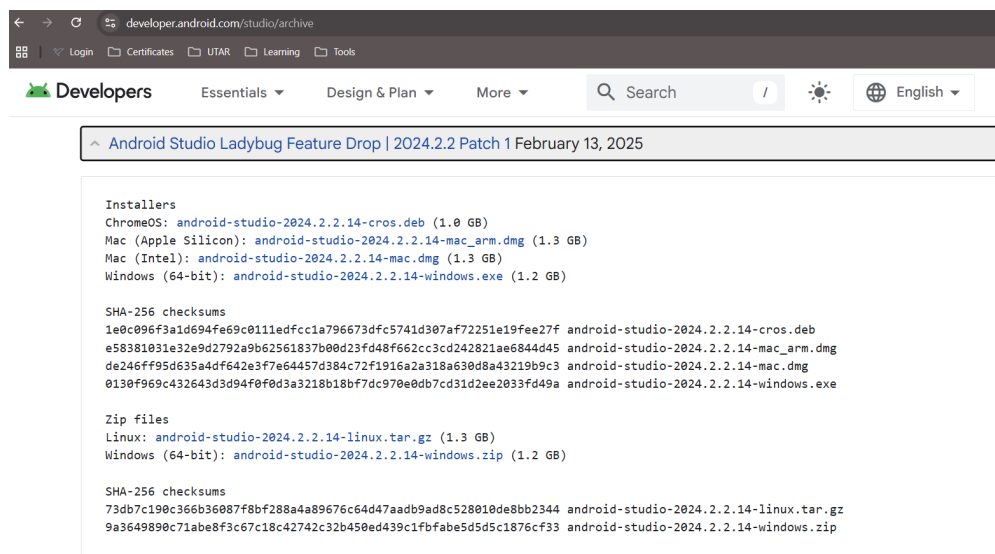


Figure 5.9 Android Studio Download Page

Aside from that, it is important to grant the necessary permission for the application to ensure it will be able to perform as expected.

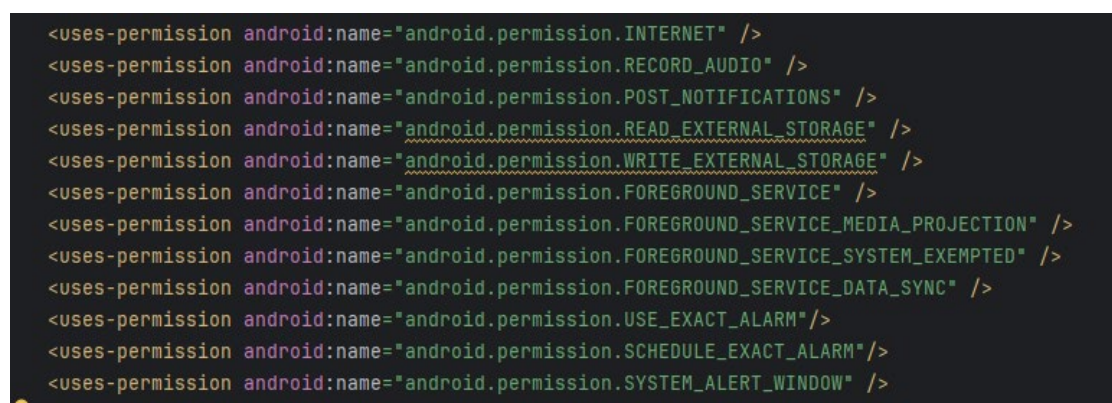


Figure 5.10 Android Studio Android Manifest XML File

CHAPTER 5

Furthermore, the application development will also require additional dependencies to integrate more libraries that are required for the development.

```
//gif
implementation("pl.droidsonroids.gif:android-gif-drawable:1.2.25")

//navigation bar
implementation("com.google.android.material:material:1.3.0-alpha03")

//loading animation
implementation("com.github.Marvel999:Android-Loading-Animation:1.0.0")

//Gemini
implementation("com.google.ai.client.generativeai:generativeai:0.7.0")
implementation("com.google.guava:guava:31.0.1-android")
implementation("org.reactivestreams:reactive-streams:1.0.4")

//OCR google ml kit
implementation("com.google.firebase:firebase-ml-vision:24.0.3")
implementation("com.google.firebase:firebase-ml-vision-image-label-model:20.0.1")
implementation("com.google.android.gms:play-services-mlkit-text-recognition:18.0.0")
implementation("com.google.mlkit:text-recognition:16.0.0")
```

Figure 5.11 Android Studio Build Grandle Kotlin File Dependencies Part I

```
//Okhttp
implementation("com.squareup.okhttp3:okhttp:4.12.0")

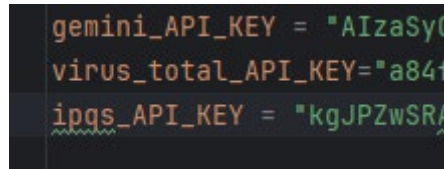
implementation("androidx.appcompat:appcompat:1.6.1")
implementation("com.google.android.material:material:1.11.0")
implementation("androidx.constraintlayout:constraintlayout:2.1.4")
testImplementation("junit:junit:4.13.2")
androidTestImplementation("androidx.test.ext:junit:1.1.5")
androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")

//QR
implementation("com.journeyapps:zxing-android-embedded:4.3.0")
implementation("com.google.zxing:core:3.2.0")

}}
dependencies {
    implementation("androidx.appcompat:appcompat:1.6.1")
    implementation("com.google.android.material:material:1.11.0")
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")
    implementation("androidx.activity:activity:1.8.0")
}
```

Figure 5.12 Android Studio Build Grandle Kotlin File Dependencies Part II

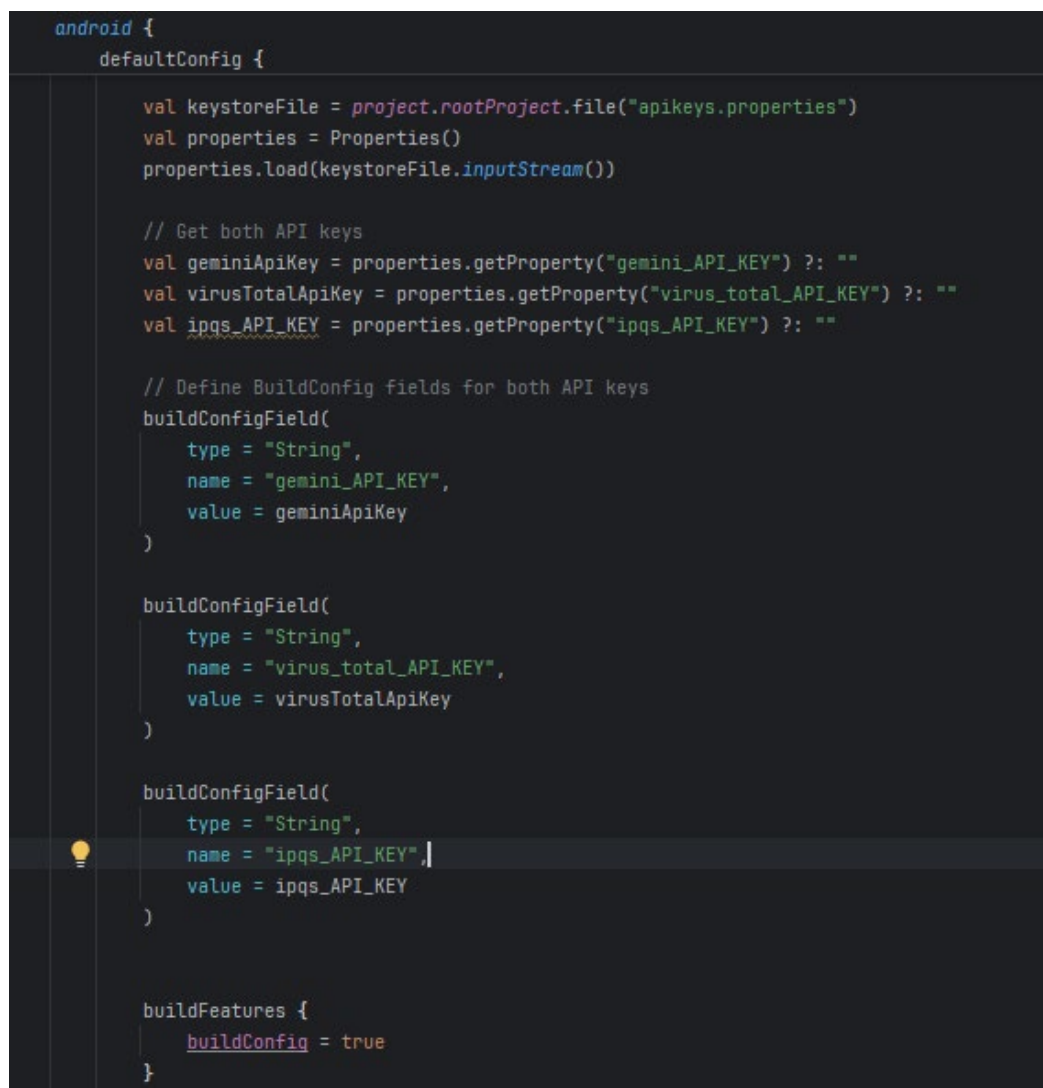
Then lastly, later on the API keys obtained from the services that the project is using should be stored under the root directory of the project folder. Firstly, a file named “apikey.properties” is created with all the API Key declared.



```
gemini_API_KEY = "AIzaSyC
virus_total_API_KEY="a84f
ipqs_API_KEY = "kgJPZwSRA
```

Figure 5.13 API Keys Properties File

Last build gradle file, the following configuration can be seen within the figure shown. Since we have a dedicated file for the API keys, we are able to exclude the API key folder when uploading to GitHub in the future.



```
android {
    defaultConfig {

        val keystoreFile = project.rootProject.file("apikey.properties")
        val properties = Properties()
        properties.load(keystoreFile.inputStream())

        // Get both API keys
        val geminiApiKey = properties.getProperty("gemini_API_KEY") ?: ""
        val virusTotalApiKey = properties.getProperty("virus_total_API_KEY") ?: ""
        val ipqs_API_KEY = properties.getProperty("ipqs_API_KEY") ?: ""

        // Define BuildConfig fields for both API keys
        buildConfigField(
            type = "String",
            name = "gemini_API_KEY",
            value = geminiApiKey
        )

        buildConfigField(
            type = "String",
            name = "virus_total_API_KEY",
            value = virusTotalApiKey
        )

        buildConfigField(
            type = "String",
            name = "ipqs_API_KEY",
            value = ipqs_API_KEY
        )

        buildFeatures {
            buildConfig = true
        }
    }
}
```

Figure 5.14 Android Studio Setting Up API Keys

5.2.2 Google AI studio

Google AI studio will provide us with the Google's chatbot Gemini API key for later on development. It should be noted that this service does not require the user to download any software to obtain the API key. Firstly, go to https://aistudio.google.com/prompts/new_chat and register a new account if you do not have one. Then click agree to their terms and conditions.

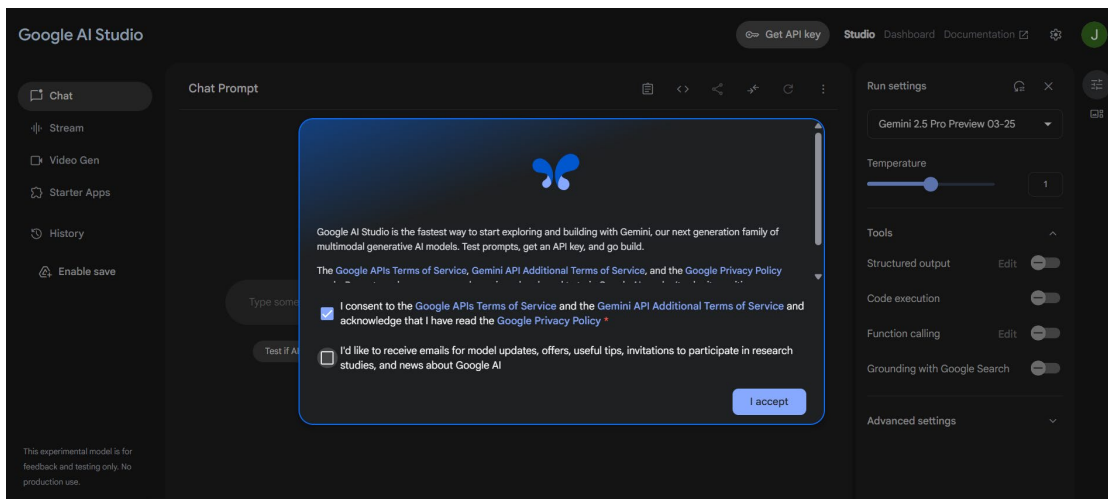


Figure 5.15 Google AI Studio Webpage

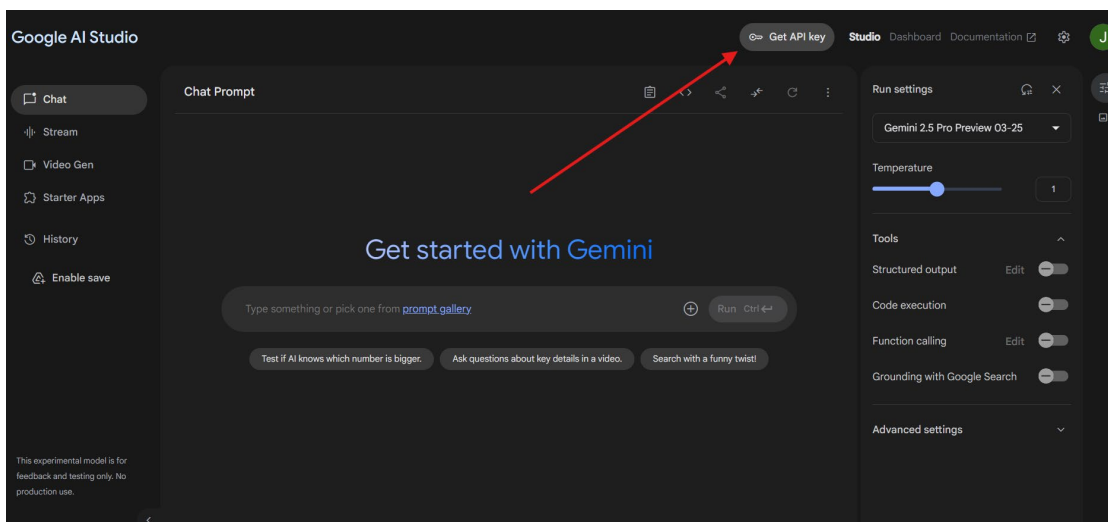


Figure 5.16 Google AI Studio Navigate to API Key Page

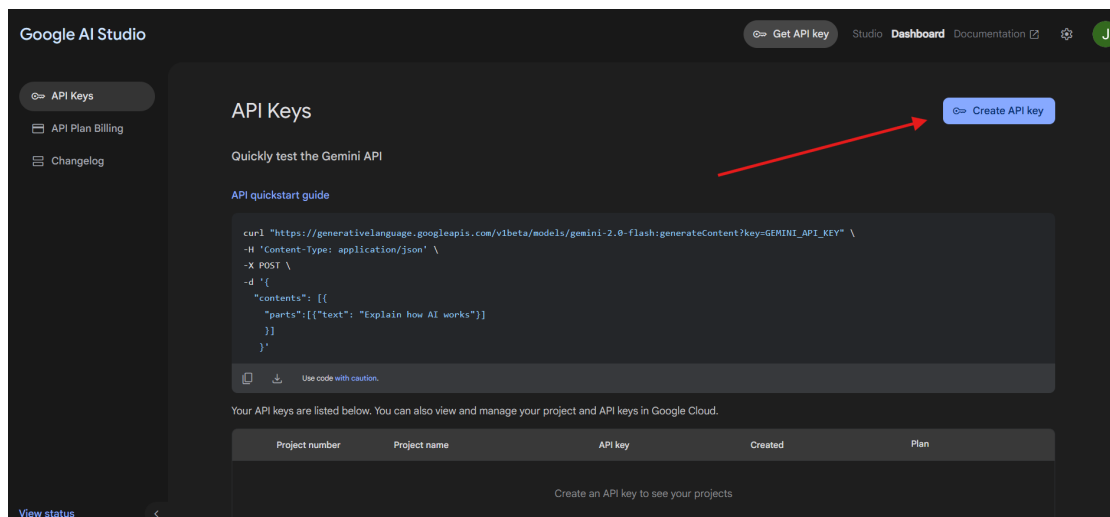


Figure 5.17 Google AI Studio Create API Key

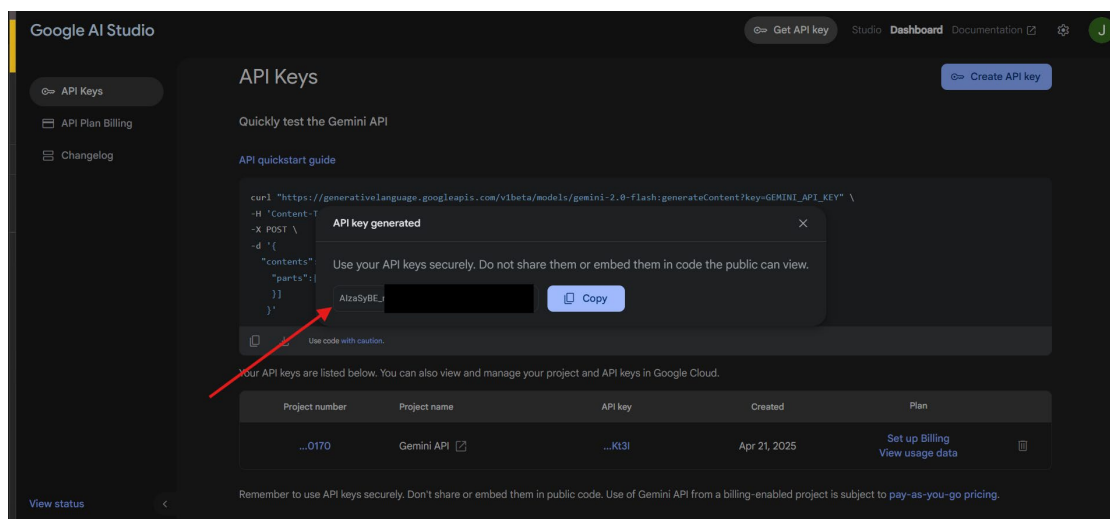


Figure 5.18 Google AI Studio Retrieve Gemini Key

Based on the steps shown in Figure 5.15 to Figure 5.18, you will be able to obtain your own Gemini API Key that you can use in your project for free. All the popular and recent Gemini models are available to use free of charge. Although the free tier has been imposed with some limitation such as limited quota per day, it does not impact the project and the throughout the development of the application, the project's usage on the Gemini API Key rarely finishes the quota given. It should be noted that the Gemini model that we have used for this project is Gemini 2.0 Flash Thinking, as it is sufficient for our use case.

5.2.3 IP Quality Score

IP Quality Score (IPQS) will become the module for the application to have the ability to scan website links for any malicious activity. The following will be the guide to obtain the API key of the service.

Figure 5.19 IPQS Login Page

The user may login to the IPQS website via registering a new account or login via Google account.

Figure 5.20 IPQS Registration Page

The website may require further information from the user to complete the free account registration.

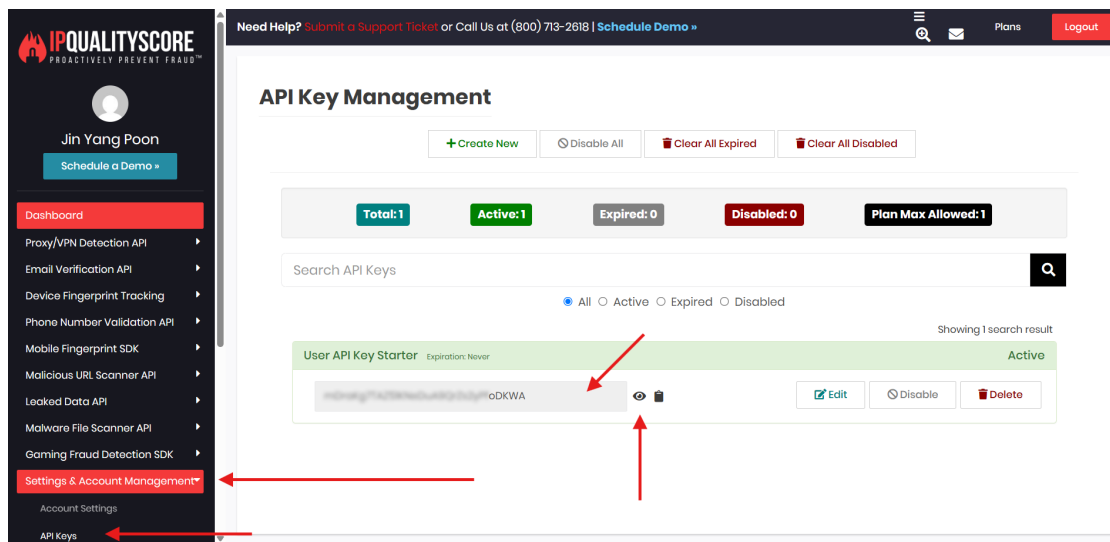


Figure 5.21 IPQS Navigating to API Key

The user will be able to obtain the IPQS API key by navigating to the side panel on the left, choosing “Settings & Account Management” and “API Keys”. Then the user will be able to view the API key that is already created for every user. The user may click on the eye icon to show the API key.

5.3 Integrating Services and Coding

Since the complete application consists of 34 classes with each class on average containing 300 lines of code, this section will only prioritize displaying and explaining codes that are built to integrate the core services. These services are listed as below:

- a. Speech-To-Text
- b. Gemini API Integration
- c. IPQS API Integration
- d. Google Machine Learning Kit Optical Character Recognition
- e. Secure Browse

5.3.1 Speech-To-Text

The built-in system or class in Android is the *SpeechRecognitionListener*, where it converts audio to text. The components of the *SpeechRecognitionListener* consist of the following [46]:

1. SpeechRecognizer

The core component that manages the speech recognition process. The *SpeechRecognizer* will listen to the audio input and convert the conversation to text.

2. RecognitionListener Interface

The *RecognitionListener* Interface will provide a set of callback methods. These methods will inform the application regarding the speech recognition process.

3. Callbacks

There are many callbacks in this class, but the most common and frequently used in developing the project's application are

- **onReadyForSpeech(Bundle params):** Called when the recognizer is ready to begin listening.
- **onBeginningOfSpeech():** Indicates that the user has started speaking.
- **onEndOfSpeech():** Indicates that the user has stopped speaking.
- **onError(int error):** Notifies about errors that occurred during recognition.
- **onResults(Bundle results):** This method is called when recognition results are available. This is where the recognized text can be obtained.
- **onPartialResults(Bundle partialResults):** This method is called when partial results are available, useful for giving immediate feedback to the user.

4. Intent Configuration

To start the speech recognition process, an *Intent* object will be created with parameters such as the language model, the prompt to display, and any other necessary settings for recognition. The recognizer will start to listen for speech based on this intent.

CHAPTER 5

It is to be noted that, the Speech-To-Text configuration in the project is to only send the recognized text after accumulated enough words, for example 10 words, only then it will send to Gemini for analysis. This is to ensure the amount of words sent to the Gemini is meaningful enough to let the chatbot make valuable decisions.

After implementing the SpeechRecognitionListener class in the prototype, the result can be shown in the following figures. It is noted that, in the prototype application configuration, after the amount of word has been recognized, only then it will be sent to Gemini for processing.

```
E hello this is Alice calling the
D Recognized text in buffer 11 : hello this is Alice calling the
E underwriting department regarding your
D Recognized text in buffer 15 : hello this is Alice calling the underwriting department regarding your
E discover credit card account based on
D Recognized text in buffer 21 : hello this is Alice calling the underwriting department regarding your discover credit
E your recent payment activity and balance
D Recognized text in buffer 27 : hello this is Alice calling the underwriting department regarding your discover credit
E you are eligible for an interest rate
D Recognized text in buffer 34 : hello this is Alice calling the underwriting department regarding your discover credit
```

Figure 5.22 Audio Capture in Process

```
I Gemini called and working on it
I Gemini will be processing this context
hello this is Alice calling the underwriting department regarding your discover credit card account based on your recent
payment activity and balance you are eligible for an interest rate reduction to as low as 1.9% to take
```

Figure 5.23 Gemini Scanning the Conversation

```
E Gemini scan result : Output: **Dangerous**

Description: This message is a potential phishing attempt. It claims to be from the "underwriting department" regarding a
Discover credit card account, offering a low interest rate reduction. This type of message is commonly used by scammers to
lure individuals into providing sensitive personal information, such as credit card details, account numbers, or social
security numbers. The message lacks any clear identification of the sender or specific details about the account, making it
highly suspicious. It's important to remember that legitimate financial institutions would never ask for sensitive
information through unsolicited phone calls or messages.
I Gemini done scanning
```

Figure 5.24 Gemini Result

Gemini Input:

“hello this is Alice calling the underwriting department regarding your discover credit card account based on your recent payment activity and balance you are eligible for an interest rate reduction to as low as 1.9% to take”

Gemini Result : Output: ****Dangerous****

Description: This message is a potential phishing attempt. It claims to be from the "underwriting department" regarding a Discover credit card account, offering a low interest rate reduction. This type of message is commonly used by scammers to lure individuals into providing sensitive personal information, such as credit card details, account numbers, or social security numbers. The message lacks any clear identification of the sender or specific details about the account, making it highly suspicious. It's important to remember that legitimate financial institutions would never ask for sensitive information through unsolicited phone calls or messages.

Overall, the success of these test conducted on the prototype has provided the confidence on building a prototype that combines all of these features and ultimately develop a mobile application that was suggested in this project to counter scam.

5.3.2 Gemini API Integration

The following is the code for the Gemini API class; to make it more modular for the rest of the Java classes to implement the Gemini model, the best possible approach is to make it a class instead of rewriting the entire code for each time any of the Java classes requires decision making.

```
private static final String TAG = "chatbot_gemini";

1 usage
private static final SafetySetting harassmentSafety = new SafetySetting(HarmCategory.HARASSMENT,
    BlockThreshold.NONE);

1 usage
private static final SafetySetting hateSpeechSafety = new SafetySetting(HarmCategory.HATE_SPEECH,
    BlockThreshold.NONE);

1 usage
private static final SafetySetting dangerousSafety = new SafetySetting(HarmCategory.DANGEROUS_CONTENT,
    BlockThreshold.NONE);

1 usage
private static final SafetySetting sexuallySafety = new SafetySetting(HarmCategory.SEXUALLY_EXPLICIT,
    BlockThreshold.NONE);

1 usage
private static GenerativeModel gm = new GenerativeModel(
    modelName: "gemini-2.0-flash",
    BuildConfig.gemini_API_KEY,
    generationConfig: null,
    Arrays.asList(
        harassmentSafety,
        hateSpeechSafety,
        dangerousSafety,
        sexuallySafety
    )
);

1 usage
private static GenerativeModelFutures model = GenerativeModelFutures.from(gm);
1 usage
```

Figure 5.25 Gemini API Model Code Configuration

In the Figure 5.25, we are able to observe the configuration for the Gemini model. The four settings which includes harassment, hate speech, dangerous, sexually safety should be set to not block the content. This is because the project aims to deploy an application that will analyze and return whether the content is safe or otherwise, if these settings are not disabled, the generative model will not provide any valuable feedback once malicious activity is detected. Additionally, the Gemini variant used in this project is the Gemini 2.0 Flash version. This version of the model is able to produce the result with a decent accuracy and does not take too much of the limited quota. It is to be noted that, this project has not implemented any custom trained variant of the Gemini model, the model being used has been sent with a detailed prompt amend before the content that is required to be scanned. The block of text can be seen at the following


```
private static String setup = "Input:\n" +
    "The text is captured from a screenshot and converted to text. Accuracy may vary.\n" +
    "If the input is HTML, the LLM will scan for malicious code before proceeding with text analysis.\n" +
    "Do not use URLs for decision-making in threat assessment, but list only URLs that require scanning.\n" +
    "URLs used solely for loading images, stylesheets, fonts, or other static resources should be ignored.\n" +
    "Analyze the text for signs of phishing, scams, or mentions of known malicious software.\n" +
    "\n" +
    "Processing Steps:\n" +
    "1. Extract and list only URLs that require scanning (e.g., links directing to login pages, transactional pages, or user-interaction sites).\n" +
    "2. If the input is HTML, check for malicious code before further analysis.\n" +
    "3. Assess the text's context and likelihood of events to determine validity.\n" +
    "\n" +
    "Output Format (JSON):\n" +
    "{\n" +
    "  \"valid_urls\": [\"url1\", \"url2\", ...],\n" +
    "  \"result\": \"Safe\" | \"Dangerous\",\n" +
    "  \"description\": \"Reason for classification.\"\n" +
    "}\n" +
    "\n" +
    "Guidelines:\n" +
    "Safe Environment:\n" +
    "- \"result\": \"Safe\"\n" +
    "- \"description\": Brief explanation of why the environment is safe.\n" +
    "- If the event is rare but possible, mention its rarity.\n" +
    "\n" +
    "Dangerous Environment:\n" +
    "- \"result\": \"Dangerous\"\n" +
    "- \"description\": Detailed reasoning highlighting specific threats, such as:\n" +
    "  - Phishing attempts (e.g., fake login pages, requests for sensitive information).\n" +
    "  - Scam-related language (e.g., lottery winnings, urgent payment requests).\n" +
    "  - Malicious software mentions (e.g., trojans, ransomware names).\n" +
    "\n" +
    "Specific Considerations:\n" +
    "- News Validity: Only flag news as dangerous if verified to be fake.\n" +
    "- Likelihood of Events: If an event is impossible, flag it as fake. Otherwise, determine if it's rare but possible.\n" +
    "- HTML Input: If the scanned HTML does not contain \"Danger\", it is considered safe, and text analysis proceeds.\n" +
    "\n" +
    "Example Outputs:\n" +
    "Safe Scenario:\n" +
    "{\n" +
    "  \"valid_urls\": [\"https://example.com/login\"],\n" +
    "  \"result\": \"Safe\",\n" +
    "  \"description\": \"The login page is verified as safe. No suspicious elements detected.\"\n" +
    "}\n" +
    "\n" +
    "Dangerous Scenario:\n" +
    "{\n" +
    "  \"valid_urls\": [\"https://phishing-site.com\", \"https://scam-alert.com\"],\n" +
    "  \"result\": \"Dangerous\",\n" +
    "  \"description\": \"A phishing attempt has been detected. The message contains a fake login request and suspicious links.\"\n" +
    "}";
```

Figure 5.26 Gemini Instruction Part I

```
"Dangerous Environment:\n" +
"- \"result\": \"Dangerous\"\n" +
"- \"description\": Detailed reasoning highlighting specific threats, such as:\n" +
  - Phishing attempts (e.g., fake login pages, requests for sensitive information).\n" +
  - Scam-related language (e.g., lottery winnings, urgent payment requests).\n" +
  - Malicious software mentions (e.g., trojans, ransomware names).\n" +
"\n" +
"Specific Considerations:\n" +
"- News Validity: Only flag news as dangerous if verified to be fake.\n" +
"- Likelihood of Events: If an event is impossible, flag it as fake. Otherwise, determine if it's rare but possible.\n" +
"- HTML Input: If the scanned HTML does not contain \"Danger\", it is considered safe, and text analysis proceeds.\n" +
"\n" +
"Example Outputs:\n" +
"Safe Scenario:\n" +
"{\n" +
  \"valid_urls\": [\"https://example.com/login\"],\n" +
  \"result\": \"Safe\",\n" +
  \"description\": \"The login page is verified as safe. No suspicious elements detected.\"\n" +
"}\n" +
"\n" +
"Dangerous Scenario:\n" +
"{\n" +
  \"valid_urls\": [\"https://phishing-site.com\", \"https://scam-alert.com\"],\n" +
  \"result\": \"Dangerous\",\n" +
  \"description\": \"A phishing attempt has been detected. The message contains a fake login request and suspicious links.\"\n" +
"}";
```

Figure 5.27 Gemini Instruction Part II

In figure 5.26 and 5.27 we are able to observe the instruction given to generative model to ensure consistent and valuable output is provided to the user. It is well noted that fine-tuning the model is an option; however, this project has not taken that approach instead opted for the in-context learning method. The in-content learning (ICL) is a technique where the instructions and demonstrations are placed into the prompt in a natural language format [47]. Natural language format refers to the way that human naturally use communicate when speaking and writing to each other.

```

4 usages 3 implementations
public interface AnalyzeCallback{
    1 usage 3 implementations
    void onResult(String text);
}

4 usages
public String analyze(String information, AnalyzeCallback callback) {
    Content content = new Content.Builder()
        .addText(setup + information)
        .build();
    Executor executor = Executors.newSingleThreadExecutor();
    ListenableFuture<GenerateContentResponse> response = model.generateContent(content);
    Log.i(TAG, msg: "Gemini called and working on it");
    Log.i(TAG, msg: "Gemini will be processing this context\n"+information);
    Futures.addCallback(response, new FutureCallback<GenerateContentResponse>() {
        @Override
        public void onSuccess(GenerateContentResponse result) {
            String resultText = result.getText();
            if (callback != null) {
                resultText = resultText.replace(target: "```json", replacement: "");
                resultText = resultText.replace(target: "```", replacement: "");
                callback.onResult(resultText);
                Log.i(TAG, msg: "Gemini done scanning");
            }
        }
        @Override
        public void onFailure(Throwable t) {
            // Handle failure
            Log.e(TAG, "Gemini has failed to scan information, reason : "+t.toString());
        }
    }, executor);

    // Doesn't return a meaningful value, might return null or empty string
    return "";
}

```

Figure 5.28 Gemini Result Implementation

In Figure 5.28, the design to handle the Gemini API is straight forward. The `AnalyzeCallback` interface will be a callback mechanism to deliver the result asynchronously. While the `analyze` method will build up the prompt, assign a thread to perform the work, then call the API and send the setup prompt and the content prompt to be analyzed. Once finish, the callback method will be invoked and the result will be sent to the interface method.

5.3.3 IPQS API Integration

The following figure contains the code regarding the IPQS class module.

```

84 private class MaliciousUrlScannerTask extends AsyncTask<String, Void, String> {
    no usages
86     @Override
87     protected String doInBackground(String... params) {
88         String url = params[0];
89         int strictness = Integer.parseInt(params[1]);
90
91         try {
92             // Construct the API URL with proper encoding
93             String apiUrl = "https://www.ipqualityscore.com/api/json/url/" +
94                 API_KEY + "/" + Uri.encode(url) + "?strictness=" + strictness;
95
96             // Make the API request
97             URL apiUrl = new URL(apiUrl);
98             HttpURLConnection urlConnection = (HttpURLConnection) apiUrl.openConnection();
99             try {
100                 InputStream in = urlConnection.getInputStream();
101                 Scanner scanner = new Scanner(in);
102                 scanner.useDelimiter("pattern: \"\\A\"");
103                 boolean hasInput = scanner.hasNext();
104                 if (hasInput) {
105                     String response = scanner.next();
106                     return readWebResult(new JSONObject(response));
107                 }
108             } finally {
109                 urlConnection.disconnect();
110             }
111         } catch (IOException | JSONException e) {
112             e.printStackTrace();
113             return "Error occurred: " + e.getMessage();
114         }
115
116         return "Unknown Error";
117     }
118 }
119
120 }

```

Figure 5.29 IPQS Calling API Code

Figure 5.29 shows a method in the IPQS class called MaliciousUrlScannerTask. This class will be responsible for calling the API from the IPQS service. It will make a HTTP request and take in the JSON format response from the IPQS scan result.

```

public String readWebResult(JSONObject result){
    String reply="Result\n";
    try {
        if (!result.getBoolean( name: "success")) {
            Log.e("URLCHECKER : ", "Failed to sent info to website");
            reply+= "ERR : FAILED TO SEND";
        }
        if (result.getInt( name: "status_code") != 200) {
            Log.e("URLCHECKER : ", "Website failure" + result.getString( name: "status_code"));
            reply+= "ERR : status code " + result.getString( name: "status_code")+"\n";
        }
    }

    if(result.getBoolean( name: "spamming"))
        reply+="Spamming Website Spotted\n";

    if(result.getBoolean( name: "malware"))
        reply+="Malware Website Spotted\n";

    if(result.getBoolean( name: "phishing"))
        reply+="Phishing Website Spotted\n";

    if(result.getBoolean( name: "suspicious"))
        reply+="Suspicious Website Spotted\n";

    if(result.getBoolean( name: "adult"))
        reply+="Adult Website Spotted\n";

    reply+="Risk Score : " + result.getString( name: "risk_score")+"\n Category : " +
        result.getString( name: "category")+"\n";
    }
    catch (Exception e){
        e.printStackTrace();
    }
    return result.toString();
}

```

Figure 5.30 IPQS Analyzing the JSON Response

In the Figure 5.30, we are able to see the function that is coded to convert the JSON form result to more user friendly result.

```

4 usages
public MaliciousUrlScanner(){
    MaliciousUrlScannerTask ActiveTask = new MaliciousUrlScannerTask();
}

//Methods
4 usages
public String scanTargetURL(String rawURL) {
    FutureTask<String> futureTask = new FutureTask<>(() -> {
        MaliciousUrlScannerTask scannerTask = new MaliciousUrlScannerTask();
        scannerTask.execute(...params: rawURL, "0"); // You may adjust the strictness value
        Log.i( tag: "SCAN TARGET URL : ", msg: " last line of future task met");
        return scannerTask.get();
    });

    try {
        // Run the task on the current thread
        futureTask.run();
        Log.i( tag: "FUTURE TASK : ", msg: " futureTask.run() is run ");
        return futureTask.get(); // This will block until the task is complete
    } catch (Exception e) {
        e.printStackTrace();
        return "Error occurred: " + e.getMessage();
    }
}
1 usage

```

Figure 5.31 IPQS Main Function

Figure 5.31 shows the main public function that other class can invoke to access the IPQS scan website link feature. By calling the “scanTargetURL” method and placing the website link as the parameter, the scan result will be return accordingly.

5.3.4 Google Machine Learning Kit Optical Character Recognition

The Google Machine Learning Kit OCR implementation can be seen below.

```
2 usages
public interface OCRCallBack{
    4 usages
    void onTextRecognized(String text, Exception error);
}
```

Figure 5.32 Google ML Kit OCR Callback Method

```
public String analyze(String filePath, OCRCallBack callback) throws IOException {
    InputImage image = InputImage.fromFilePath(context, Uri.fromFile(new File(filePath)));
    Task<Text> result = recognizer.process(image)
        .addOnSuccessListener(new OnSuccessListener<Text>() {
            @Override
            public void onSuccess(Text visionText) {
                StringBuilder recognizedText = new StringBuilder();
                // Extract text from blocks and lines
                for (Text.TextBlock block : visionText.getTextBlocks()) {
                    for (Text.Line line : block.getLines()) {
                        String text = line.getText();
                        recognizedText.append(text).append("\n"); // Add newline for readability
                    }
                }
                Log.e("googleOCR", "googleOCR result : "+recognizedText.toString());
                callback.onTextRecognized(recognizedText.toString(), error: null);
            }
        })
        .addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                // Handle failure
                callback.onTextRecognized(text: null, e);
            }
        });
    return null;
}
```

Figure 5.33 Google ML Kit OCR Analyze Method Variant I

```

public String analyze(Uri uri, OCRCallBack callback) throws IOException {
    InputImage image = InputImage.fromFilePath(context, uri);
    Task<Text> result = recognizer.process(image)
        .addOnSuccessListener(new OnSuccessListener<Text>() {
            @Override
            public void onSuccess(Text visionText) {
                StringBuilder recognizedText = new StringBuilder();
                // Extract text from blocks and lines
                for (Text.TextBlock block : visionText.getTextBlocks()) {
                    for (Text.Line line : block.getLines()) {
                        String text = line.getText();
                        recognizedText.append(text).append("\n"); // Add newline for readability
                    }
                }
                Log.e("googleOCR", "googleOCR result : "+recognizedText.toString());
                callback.onTextRecognized(recognizedText.toString(), error: null);
            }
        })
        .addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                // Handle failure
                callback.onTextRecognized(text: null, e);
            }
        });
    return null;
}

```

Figure 5.34 Google ML Kit OCR Analyze Method Variant II

Figure 5.32 to Figure 5.34 shows the entire code for the Google Machine Learning Kit OCR. In short, the callback method will define a contract for receiving the results of the OCR and alert the other class once the OCR image to text conversion has been completed. Then, the main function is the analyze method where there consist two variants of the method. The technique used here is overloading the method to allow two different type of parameter to be accepted. The class can invoke the method by placing the file path as a string or as a Uniform Resource Identifier (URI). This allows more flexibility on the other classes when invoking this method. The analyze class will be responsible for starting the OCR process asynchronously. Once the conversion is successful, it will extract all the recognized text block by block and invokes the callback method to send the result back to the original class that has invoked this method.

5.3.5 Secure Browse

The secure browsing class implements a secure web browser within the application. It aims to protect the user from malicious website if the user wishes to navigate and browse to a website with uncertainty. It will implement a two layer security mechanism before allowing the user to load the content of the website.

The first layer security mechanism is using the IPQS to scan the website link for any malicious history or report. Then second layer security will be loading the source code of the website to and sending to Gemini to scan for any manipulating content or any threat hidden in the website. This is because websites that are newly register may bypass certain website link scan services, but their malicious content will be caught by the second layer security mechanism. Additionally, the website will be loaded with no Javascript for an extra layer of security. Therefore, the secure browser is not intended to be used as a normal browser but only use when the user has reasons and wishes to proceed with a link that may be a threat.

As the class consist of many codes that are related to update the application's User Interface (UI), the following will only show figures of the code of the core services in this class.


```

5 usages
private void scanAndLoadUrl(String url) {
    if (url == null || url.isEmpty()) return;

    currentUrl = url;
    showLoadingState();

    // Clear previous results
    resultTextView.setText("");
    resultTextView.setVisibility(View.VISIBLE);

    // Execute URL scan in background
    executorService.execute(() -> {
        try {
            // Step 1: Check the URL with the malicious URL scanner
            performUrlScan(url);

            // Step 2: Fetch and analyze the content (runs on its own thread)
            geminiApiScan(url);
        } catch (Exception e) {
            Log.e(TAG, msg: "Error during URL scan", e);
            runOnUiThread(() -> {
                showErrorState(errorMessage: "Error scanning URL: " + e.getMessage());
            });
        }
    });
}

```

Figure 5.35 Secure Browse Scan and Load URL

```

1 usage
private void performUrlScan(String url) {
    try {
        String scanResult = maliciousUrlScanner.scanTargetURL(url);

        if (scanResult == null || scanResult.isEmpty()) {
            runOnUiThread(() -> showErrorState(errorMessage: "URL scan failed"));
            return;
        }

        databaseHelper.insertWebRecord(url, scanResult);

        // Parse the scan result and update UI
        runOnUiThread(() -> updateUiWithScanResult(scanResult));
    } catch (Exception e) {
        Log.e(TAG, msg: "Error in URL scan", e);
        runOnUiThread(() -> showErrorState(errorMessage: "URL scan error: " + e.getMessage()));
    }
}

```

Figure 5.36 Secure Browse Scanning URL Code

```

1 usage
private void geminiApiScan(String url) {
    try {
        // Fetch the web content safely
        String webContent = fetchWebContentSafely(url);

        if (webContent == null || webContent.isEmpty()) {
            runOnUiThread() -> {
                resultTextView.setText("Failed to fetch content\n\nThis may be due to:" +
                    "\n\n1. Your region is blocking this content" +
                    "\n\n2. The website prevents bot (your current browser) to load the website" +
                    "\n\nYou are not encouraged to proceed for this website, if you wish to proceed, kindly verify the identity of the person who suggested the website" +
                    " and ensure that the person does not have any malicious intention to scam or phish you");
                updateFinalStatus(isSafe: false);
            });
            return;
        }

        // Send the content to Gemini API for analysis
        geminiApi.analyze(webContent, analysisResult -> {
            runOnUiThread() -> {
                try {
                    currentText = resultTextView.getText().toString();
                    url_temp_passing = url;
                    performUrlScan2(analysisResult);
                } catch (Exception e) {
                    throw new RuntimeException(e);
                }
            });
        });
    } catch (Exception e) {
        Log.e(TAG, "Error in Gemini API scan", e);
        runOnUiThread() -> showErrorState(errorMessage: "Content analysis error: " + e.getMessage());
    }
}

```

Figure 5.37 Secure Browse Gemini API Scanning

```

1 usage
private class SecureWebViewClient extends WebViewClient {
    no usages
    @Override
    public boolean shouldOverrideUrlLoading(WebView view, WebResourceRequest request) {
        String url = request.getUrl().toString();

        // Check for potentially dangerous protocols
        if (url.startsWith("file:") || url.startsWith("javascript:") ||
            url.startsWith("data:") || url.startsWith("blob:")) {
            Log.w(TAG, "Blocked potentially dangerous URL: " + url);
            Toast.makeText(context: SecureBrowsing.this, text: "Blocked unsafe URL scheme", Toast.LENGTH_SHORT).show();
            return true;
        }

        // Scan the URL before loading
        scanAndLoadUrl(url);
        return true;
    }
}

```

Figure 5.38 Secure Browse Secure Web View Client

```

1 usage
private void updateUiWithScanResult(String scanResult) {
    try {
        JSONObject resultJson = new JSONObject(scanResult);
        StringBuilder categoryBuilder = new StringBuilder();

        // Check if any malicious tags are found
        boolean hasMaliciousTags = false;

        for (String tag : MALICIOUS_TAGS) {
            if (resultJson.optBoolean(tag, fallback: false)) {
                if (categoryBuilder.length() > 0) {
                    categoryBuilder.append("\n");
                }
                categoryBuilder.append(tag);
                hasMaliciousTags = true;
            }
        }

        // Set the result text
        String categories = categoryBuilder.length() > 0 ? categoryBuilder.toString() : "Url connection is secure\nPending AI scan";
        resultTextView.setText(categories);

        // Update UI status
        if (hasMaliciousTags) {
            statusResultIcon.setImageResource(R.drawable.wrong_icon);
            Log.d(TAG, "UpdateUiWithScanResult", msg: "This is dangerous");
            warningContainer.setVisibility(View.VISIBLE);
        } else {
            statusResultIcon.setImageResource(R.drawable.correct_icon);
        }
    } catch (JSONException e) {
        Log.e(TAG, msg: "Error parsing scan result JSON", e);
        showErrorState(errorMessage: "Error processing scan result");
    }
}

```

Figure 5.39 Secure Browse Update User Interface

```

private void performUrlScan2(String geminiResponse) {
    total_url_count=0;
    try {
        // Parse the Gemini response
        JSONObject geminiJson = new JSONObject(geminiResponse);
        JSONArray validUrls = geminiJson.getJSONArray( name: "valid_urls");
        String geminiResult = geminiJson.getString( name: "result");
        String geminiDescription = geminiJson.getString( name: "description");

        // Initialize feedback object for this scan
        urlFeedback = new SecureBrowsing.UrlFeedback();

        //get the total url to set for progressbar
        total_url_count=validUrls.length();

        // If no URLs were found, simply display Gemini's analysis
        if (validUrls.length() == 0) {
            displayScanResults(geminiResult, geminiDescription, urlFeedback);
            return;
        }
    }
}

```

Figure 5.40 Secure Browse Perform URL In Depth Part I

```

// Process each URL found by Gemini using a background thread
executorService.execute(() -> {
    try {
        for (int i = 0; i < validUrls.length(); i++) {
            try {
                String url = validUrls.getString(i);
                List<String []> temp = databaseHelper.readWebRecord(url);
                if(temp.isEmpty()){
                    processNewUrl(url);
                }else{
                    processExistingUrl(temp);
                }
            } catch (JSONException e) {
                Log.e(TAG, "Error processing URL at index " + i, e);
            }
        }
    }

    // After processing all URLs, update UI on main thread
    runOnUiThread(() -> {
        try {
            // Combine Gemini's analysis with URL scan results
            String combinedResult = determineCombinedResult(geminiResult);
            String combinedDescription = createCombinedDescription(geminiDescription);

            // Display the combined results to the user
            displayScanResults(combinedResult, combinedDescription, urlFeedback);
        } catch (Exception e) {
            Log.e(TAG, "Error displaying scan results", e);
            Toast.makeText(context, SecureBrowsing.this, "Error analyzing content", Toast.LENGTH_SHORT).show();
        }
    });
});

```

Figure 5.41 Secure Browse Perform URL In Depth Part II

The following is a list of method and description to get a better understanding of the codes and functions displayed from figure 5.35 to figure 5.41.

Table 5.2 Secure Browse Methods and Description

Function	Description
scanAndLoadUrl(String url)	This is the central orchestrator that initiates the security checks for a given URL.
performUrlScan(String url)	Interacts with the MaliciousUrlScanner to get an initial safety assessment.
geminiApiScan(String url) and performUrlScan2(String geminiResponse)	Fetches the web content and uses the GeminiAPI to perform deeper content analysis and scan URLs found within the page.

SecureWebViewClient	Crucial for intercepting and scanning URLs clicked within the WebView, preventing the loading of potentially unsafe links without prior checks.
updateUiWithScanResult(String scanResult) and displayScanResults(String result, String description, UriFeedback feedback)	Responsible for presenting the scan results to the user in a clear and informative way.

5.4 System Operation

5.4.1 Screen Scanning Activation Via Quick Access Panel

This feature is where the user will activate the screen scanning function via quick access panel while using any application.

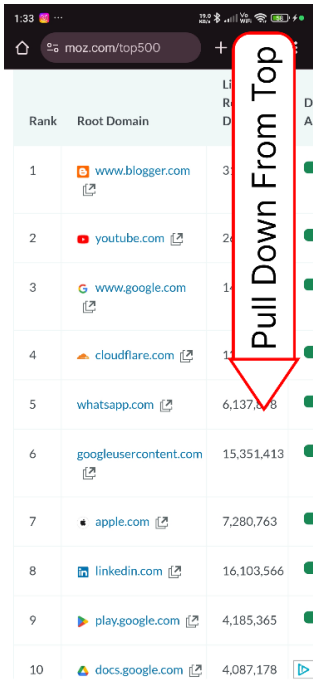


Figure 5.42 Opening Quick Access Panel

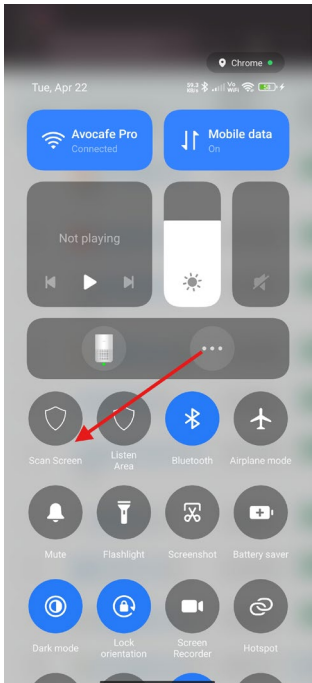


Figure 5.43 Screen Scanning Activation Via Quick Access Panel

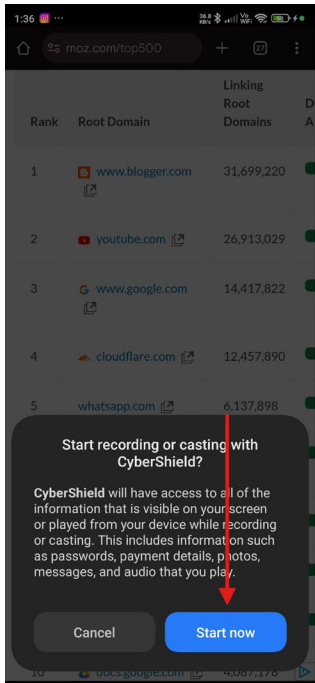


Figure 5.44 Screen Scanning Allow Application Screen Permission

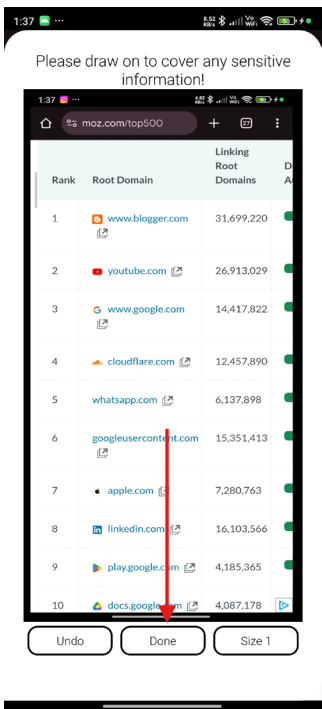


Figure 5.45 Screen Scanning Edit Page

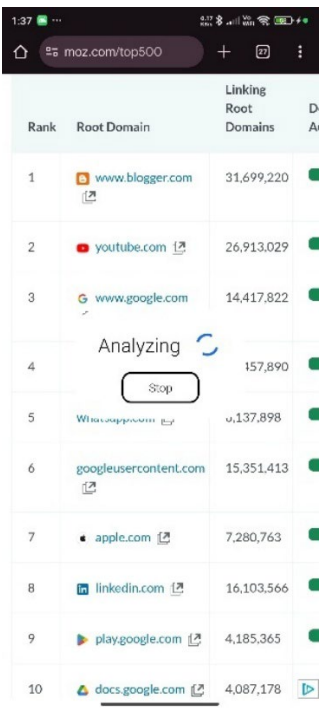


Figure 5.46 Screen Scanning Pending Result

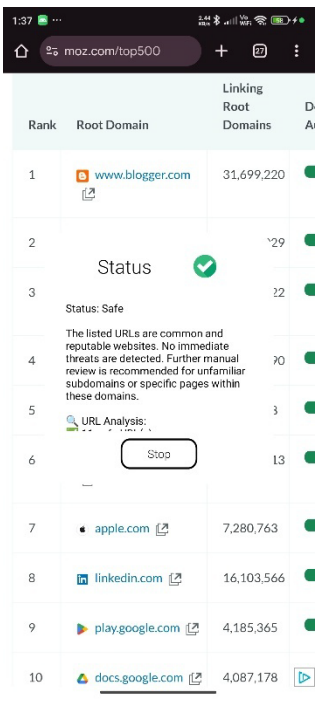


Figure 5.47 Screen Scanning Result Part I

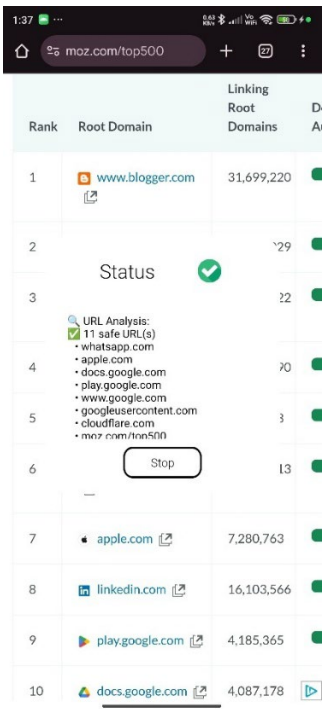


Figure 5.48 Screen Scanning Result Part II

5.4.2 Screenshot Scanning Via Image Upload inside Application

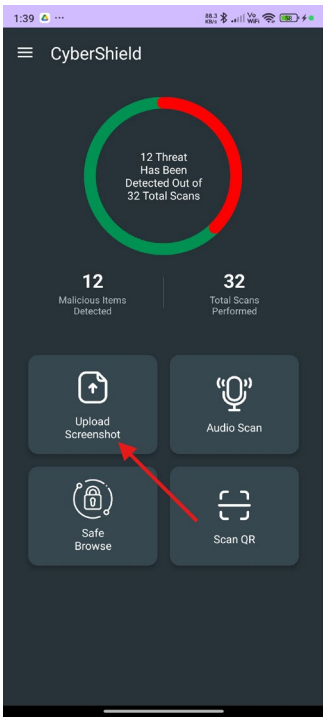


Figure 5.49 Screen Scanning Activation Inside Application

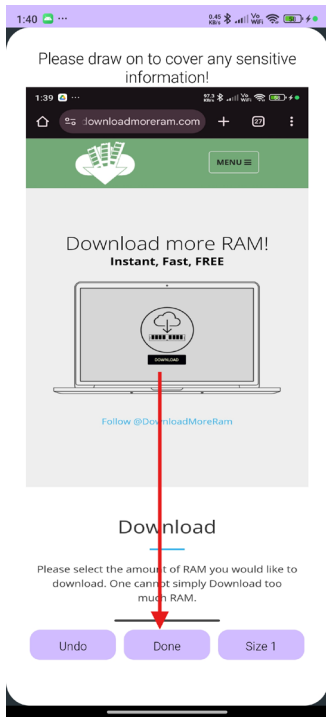


Figure 5.50 Upload Image to Screen Scanning

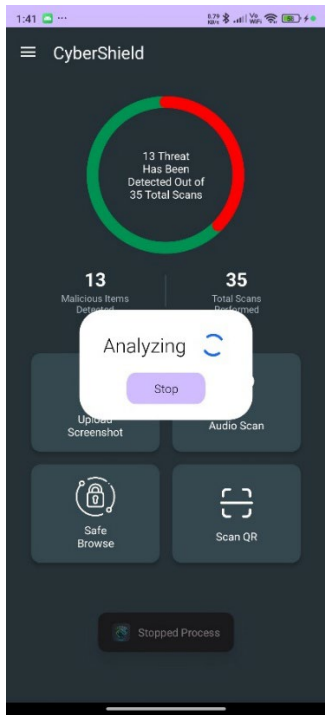


Figure 5.51 Screen Scan Analysing

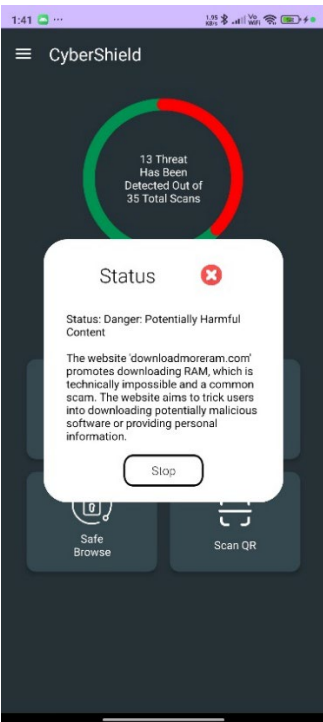


Figure 5.52 Screen Scanning Result Part I

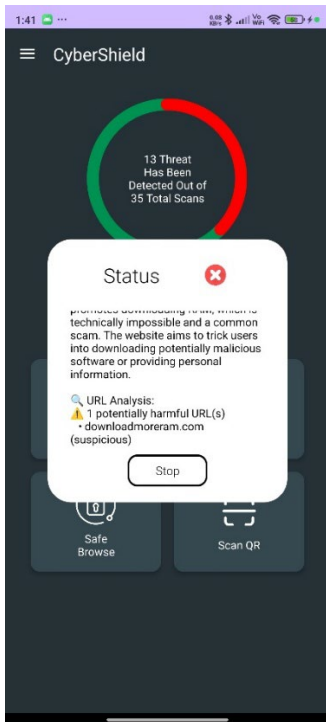


Figure 5.53 Screen Scanning Result Part II

5.4.3 Audio Scanning Activation Via Quick Access Panel

The Audio Scanning method test case will be using the social security scam voicemail available on YouTube.

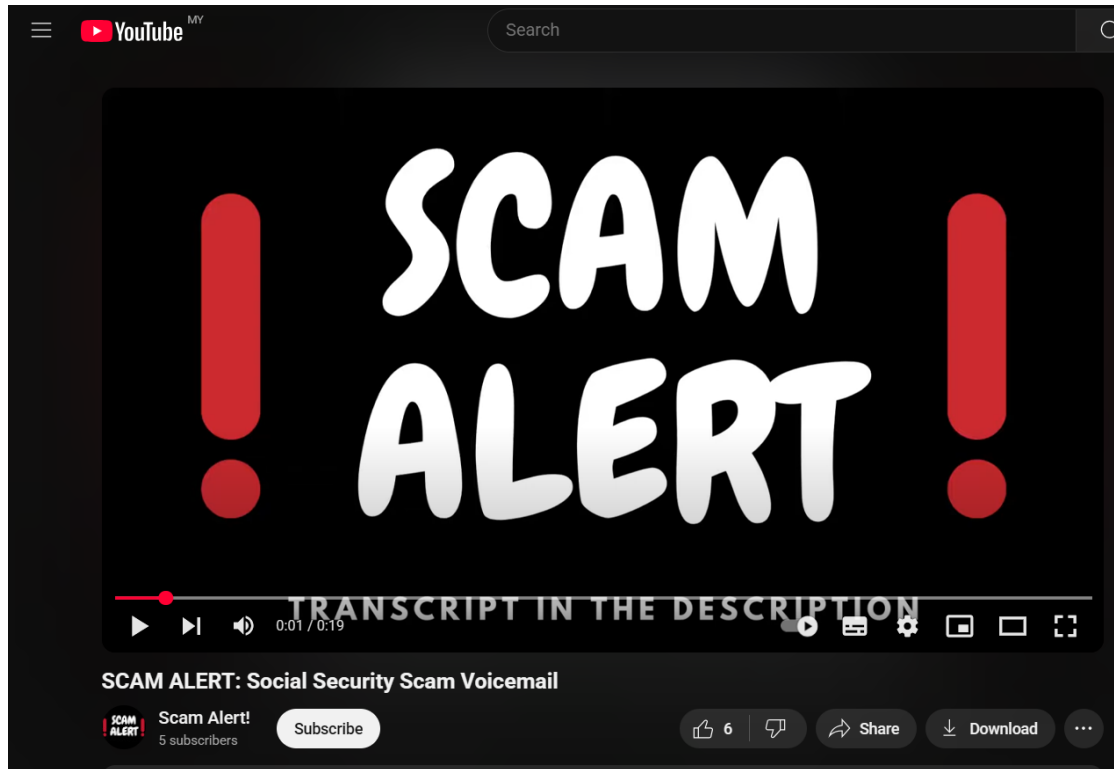


Figure 5.54 YouTube Scam Voicemail Example

The transcript is as follow “This call is from the Department of Social Security Administration. The reason you have received this phone call from our department is to inform you that we just suspend your social security number because we found some suspicious activity. So, if you want to know about this case, just press 1. Thank you.”

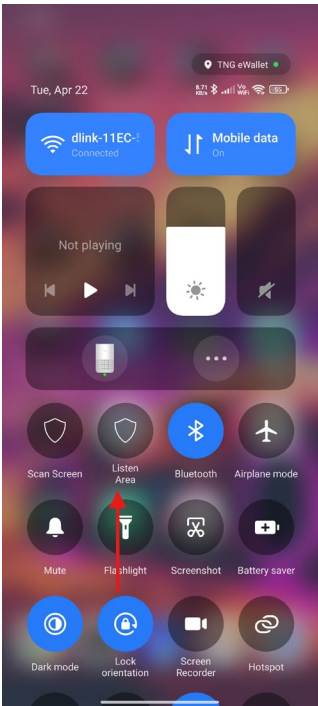


Figure 5.55 Audio Scanning
Activation at Quick Access
Panel

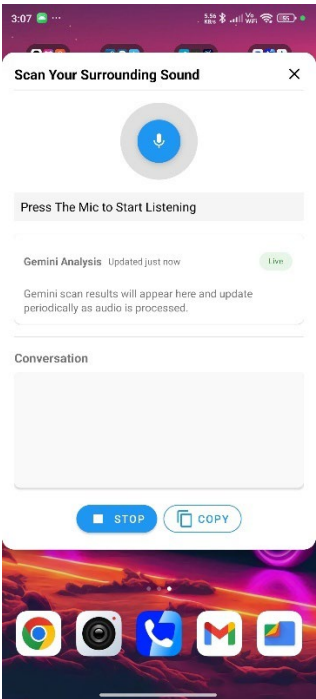


Figure 5.56 Audio Scanning
Window

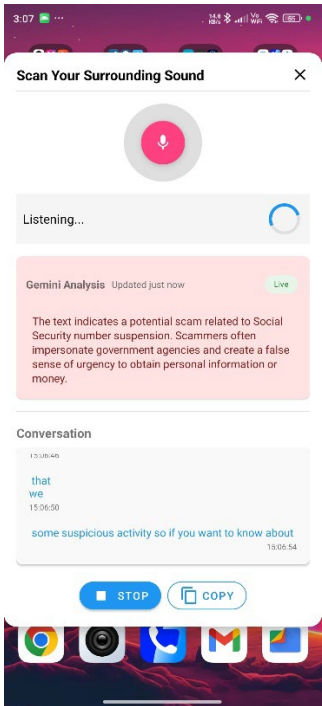


Figure 5.57 Audio Scanning
Sound Listened

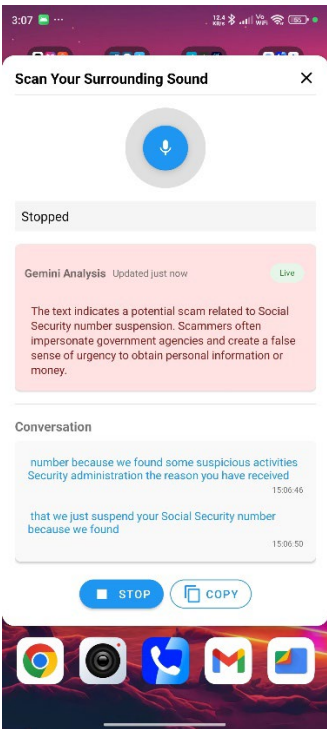


Figure 5.58 Audio Scanning
Alert

5.4.4 Safe Browsing Usage inside Application

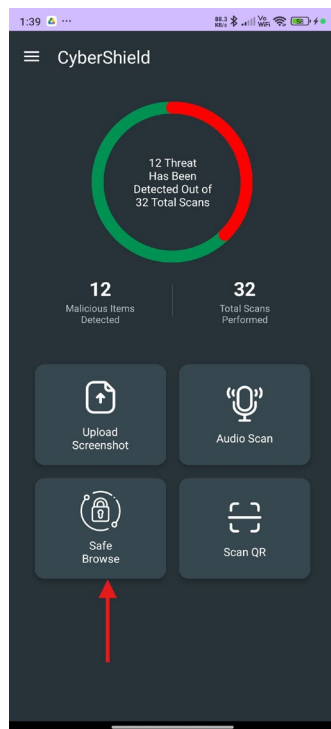


Figure 5.59 Secure Browse Activation

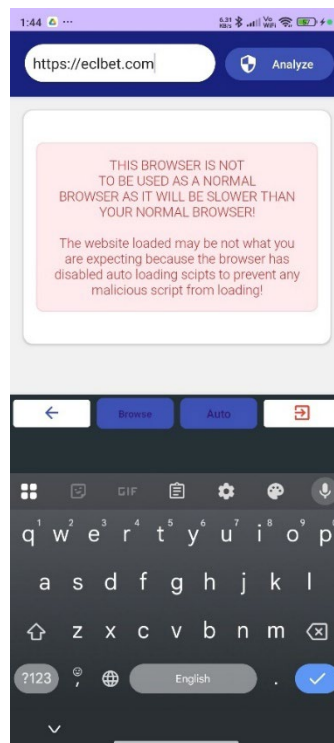


Figure 5.60 Secure Browse Page

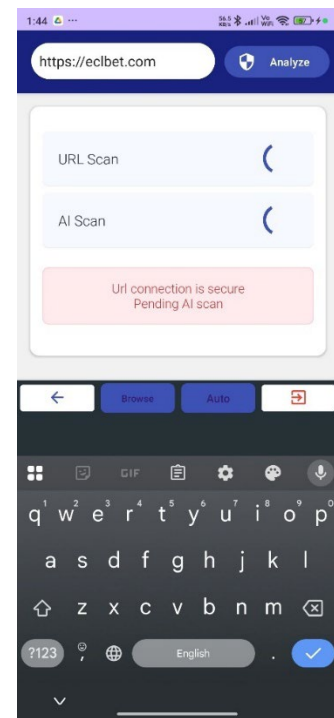


Figure 5.61 Secure Browse Analyzing

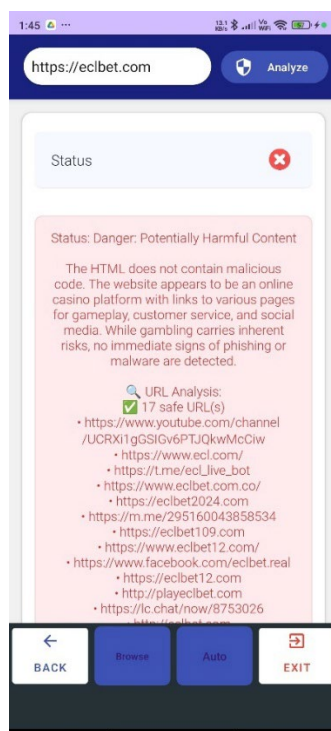


Figure 5.62 Secure Browse Threat Result

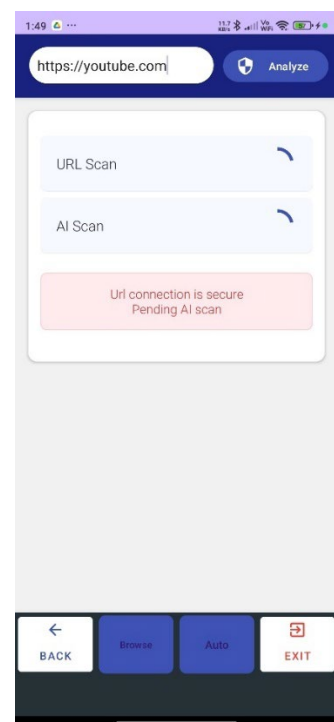


Figure 5.63 Secure Browse Scanning YouTube

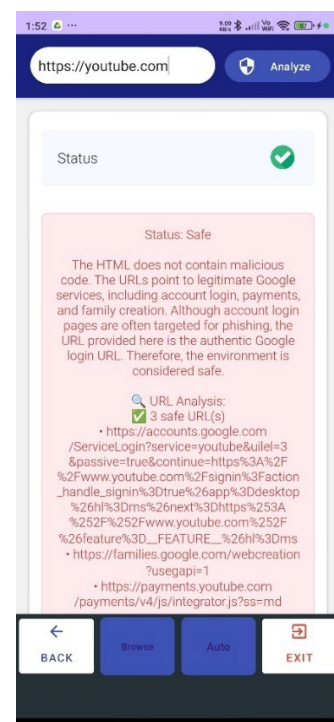


Figure 5.64 Secure Browse Safe Result

CHAPTER 5



Figure 5.65 Secure Browse
Load Website With No
Javascript

5.4.5 Scan Quick Response (QR)

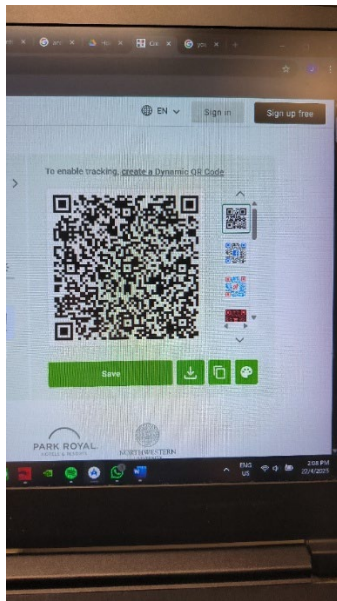


Figure 5.66 QR Sample



Figure 5.67 Scanning QR

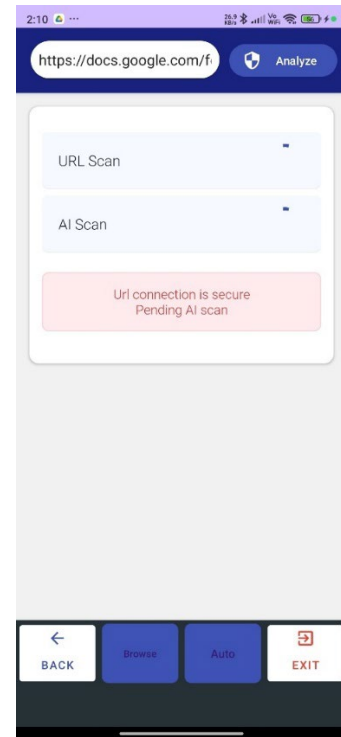


Figure 5.68 QR Load To Secure Browse

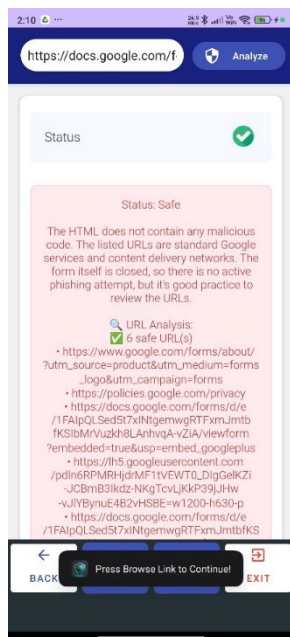


Figure 5.69 Scanning Complete Result Safe



Figure 5.70 Secure Browse Loading Website

5.4.6 Deleting Database Inside Application

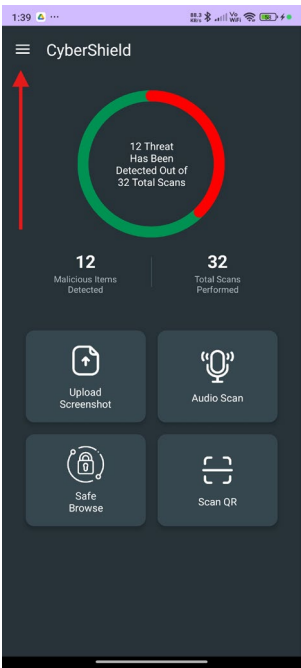


Figure 5.71 Accessing Scanned URL Database

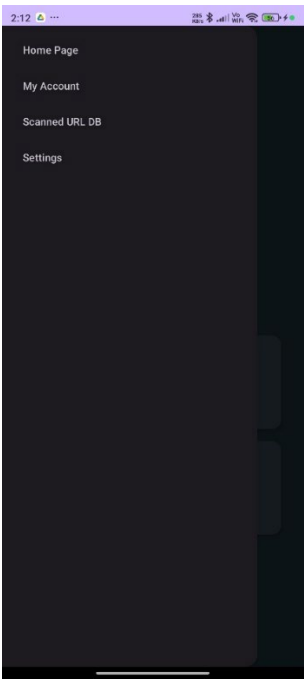


Figure 5.72 Navigating to URL Database

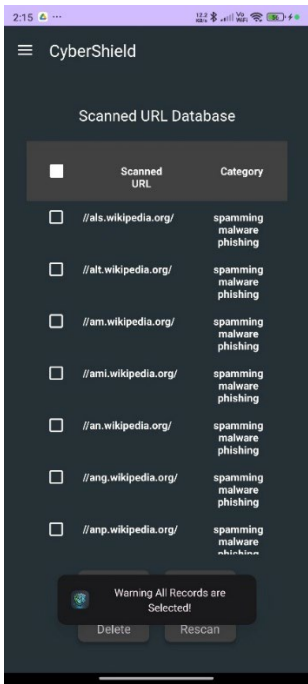


Figure 5.73 Selecting All URL

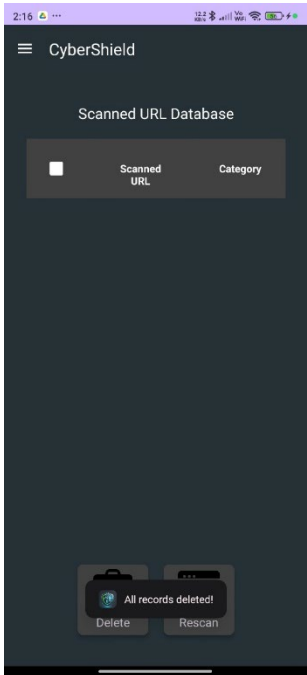


Figure 5.74 Deletion of All URL

5.5 Implementation Issues and Challenges

This section summarizes the difficulties faced when developing the application features mentioned in this project. One of the components in the screen scanning feature where the OCR functionality is being developed, the most troublesome problem faced was the accuracy of the OCR. The initial OCR model used, Tesseract, suffers very obvious accuracy problem which it would affect the later on Gemini judgement on the content that it is going to analyze. The Gemini model would constantly false flagging the content as dangerous due to the spelling of the sentence is wrong. This greatly affects its ability to accurately spot threats. The problem was fixed with the usage of Google's Machine Learning Kit OCR model where the accuracy of the image to text conversation is up to the standard and Gemini is able to properly analyze the content.

Aside from that another issue faced is that the Gemini model would require a detail prompt to suit the usage, particularly the wanted output would need a very specific JSON prompt for it to work. Manually writing the prompt for the LLM to use was tedious and does not work as consistent at most of the time. This problem was fixed by using ChatGPT, another LLM developed by Open AI, to generate the prompt for the usage to place into Gemini. The prompt generated for Gemini was specific enough for our use case and was quick and convenient. It is advised that in the future whenever a role or a particular task was needed to be assigned to an LLM, the user should ask ChatGPT or other LLM to make a prompt instead of manually typing one to ensure effectiveness.

Finally, the most challenging problem faced and has not been directly addressed is the ability of the application to listen to its phone call conversations. This is due to the nature of the Android operating system. The operating system does not simultaneously allow two application to use the audio system (microphone and speaker). Once the project's application is using the microphone, the phone conversation will not be heard from the phone speaker. Therefore, the usage of the audio scanning is not to listen to the current phone conversation or any audio but rather the concept is to deal with face to face conversation to detect any threat hidden within the other person's conversation.

5.6 Concluding Remark

In chapter 5, we are able to get a better understanding of the entire project setup including software and hardware setup. In short, the hardware setup consist of two device which is an Android with USB debugging enable and an out of the box Windows 64 bit operating system laptop. Then software setup will consist of Android Studio, Google AI Studio, IP Quality Score (IPQS) setup.

Furthermore, the chapter has also mentioned the methods with codes on integrating the core services used in the application development. This includes speech-to-text, Gemini API, IPQS API, Google Machine Learning Kit OCR, and secure browse.

In addition to the code integration, under Chapter 5.4 system operation, the demonstration of the system operation can be seen with screenshots of the application. Then finally, the implementation issues and challenges are also including in the Chapter 5.5 to state all the challenges that has been faced during the application development.

Chapter 6 System Evaluation and Discussion

6.1 System Testing and Performance Metrics

In this section, we will be performing performance benchmarking for the developed application. This includes load testing, response time analysis, and network performance. The application developed aims to offer a decent performance but still lightweight for an average performance Android smartphone.

6.1.1 Load Testing

In order to perform load testing on the application, we will utilize the built-in “Profiler”. The profiler gives the user a way to profile the app performance to better understand the cause of poor performance. The profiler will be able to let user identify areas in which the app is inefficiently using resources such as CPU, memory, graphic or even power consumption [48].

Figure 6.1 Overall CPU and Memory Usage Running Developed Application

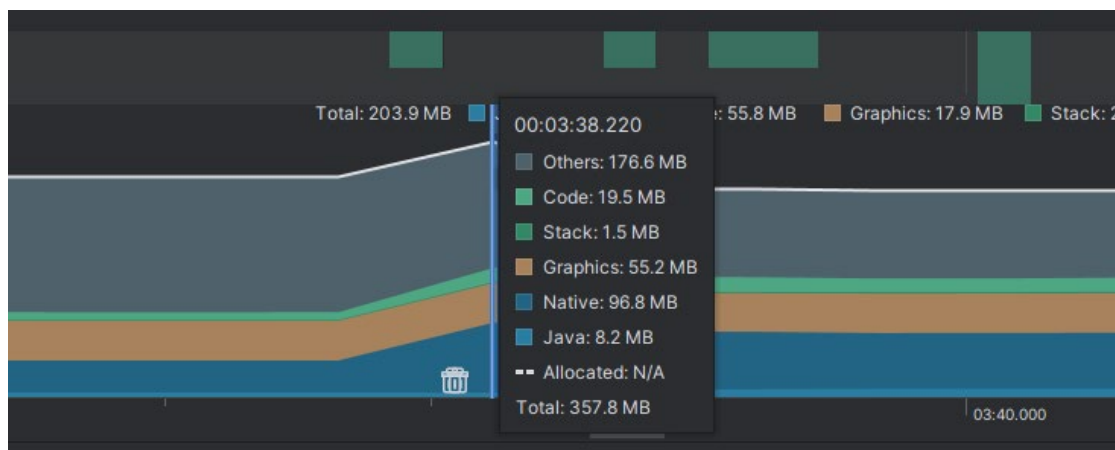


Figure 6.2 Peak Resource Usage Running Developed Application

In Figure 6.1 we can observe the overall CPU and memory usage when running and using the developed application normally. Figure 6.2 is a screenshot of the peak resource usage for the application, as show in the figure, we can see that the total

memory used is roughly 360MB of memory. The amount of memory that the application uses is insignificant.

6.1.2 Response Time

In this section, we will be measuring the response time of the application. Firstly, we will measure the response time of the screen scanning feature. To ensure accuracy of the test, the time measurement will not be manual but integrated into the source code of the application.



```
1 //After Grant Permission To Record Screen
2 long startTime = System.currentTimeMillis();
3 SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss.SSS", Locale.getDefault());
4 String formattedTime = sdf.format(new Date(startTime));
5 Log.w("Tile-ScreenScan Benchmark", "Start Time: " + formattedTime);
6
7 //At Display Pop Up Window
8 long endTime = System.currentTimeMillis();
9 SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss.SSS", Locale.getDefault());
10 String formattedTime = sdf.format(new Date(endTime));
11 Log.w("Tile-Foreground Benchmark", "End Time: " + formattedTime);
```

Figure 6.3 Code Integrated for Logging Timestamp

The following are five test case that will be screen scanned by the application to measure the response time of the screen scanning service.

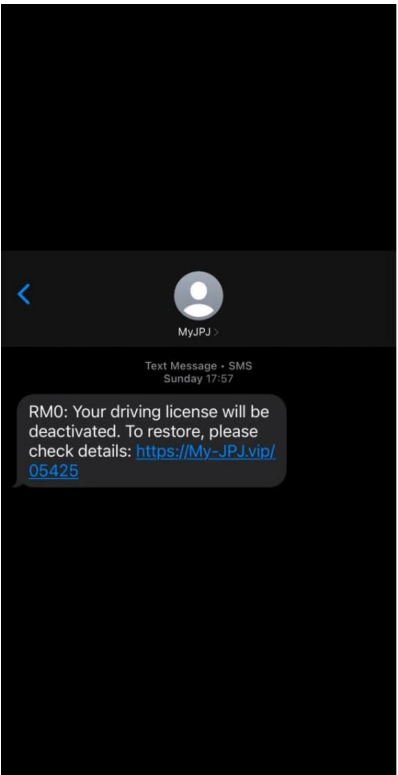


Figure 6.4 Screen Scan Test I

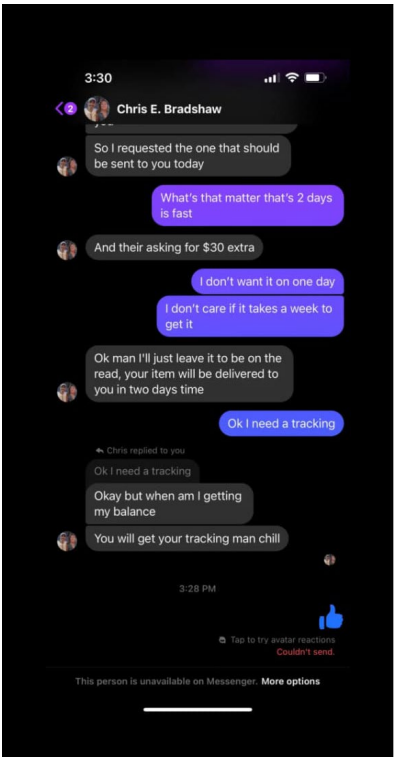


Figure 6.5 Screen Scan Test II

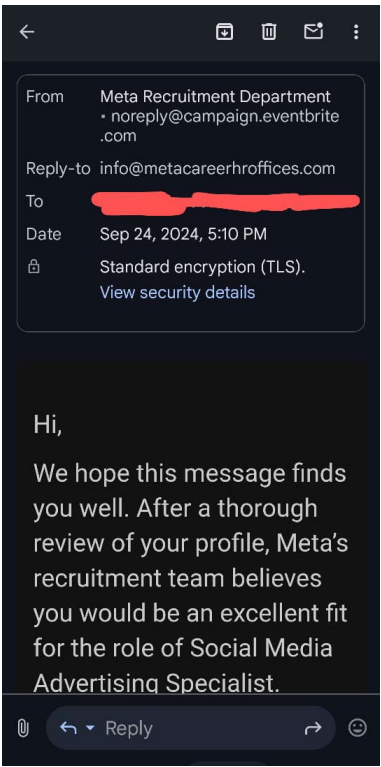


Figure 6.6 Screen Scan Test III

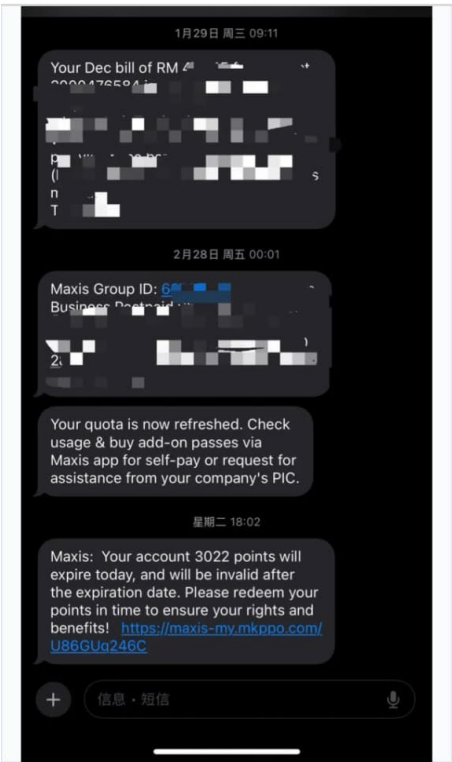


Figure 6.7 Screen Scan Test IV



Figure 6.8 Screen Scan Test V

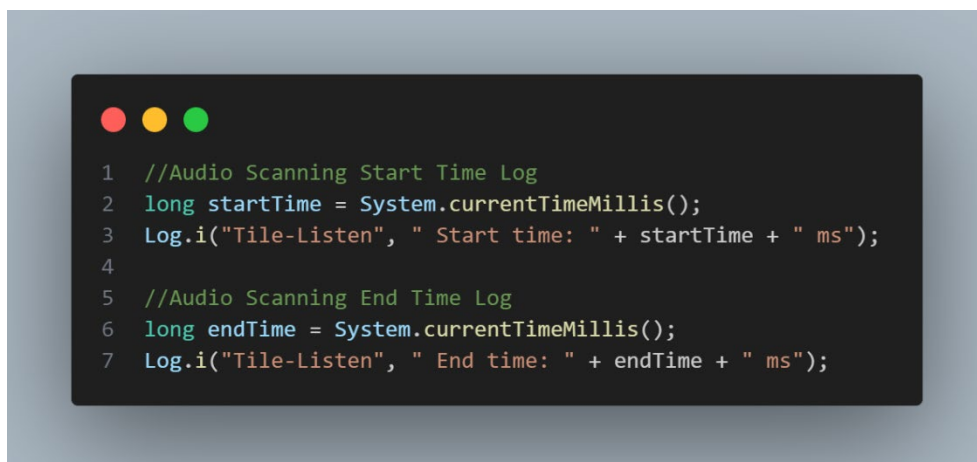
Xiaomi 22071212AG (654PYHJJQ8IV599D) Android 14										bench	
2025-04-26	19:10:23.621	20986-20986	Title-Screen...	Benchmark	com.example.fyp_ver_2					W	Start Time: 2025-04-26 19:10:23.584
2025-04-26	19:10:31.263	20986-20986	Title-Foreg...	Benchmark	com.example.fyp_ver_2					W	End Time: 2025-04-26 19:10:31.263
2025-04-26	19:11:01.939	20986-20986	Title-Screen...	Benchmark	com.example.fyp_ver_2					W	Start Time: 2025-04-26 19:11:01.933
2025-04-26	19:11:09.378	20986-20986	Title-Foreg...	Benchmark	com.example.fyp_ver_2					W	End Time: 2025-04-26 19:11:09.377
2025-04-26	19:11:15.537	20986-20986	Title-Screen...	Benchmark	com.example.fyp_ver_2					W	Start Time: 2025-04-26 19:11:15.537
2025-04-26	19:11:20.670	20986-20986	Title-Foreg...	Benchmark	com.example.fyp_ver_2					W	End Time: 2025-04-26 19:11:20.669
2025-04-26	19:11:28.071	20986-20986	Title-Screen...	Benchmark	com.example.fyp_ver_2					W	Start Time: 2025-04-26 19:11:28.064
2025-04-26	19:11:33.543	20986-20986	Title-Foreg...	Benchmark	com.example.fyp_ver_2					W	End Time: 2025-04-26 19:11:33.542
2025-04-26	19:11:40.311	20986-20986	Title-Screen...	Benchmark	com.example.fyp_ver_2					W	Start Time: 2025-04-26 19:11:40.308
2025-04-26	19:11:47.524	20986-20986	Title-Foreg...	Benchmark	com.example.fyp_ver_2					W	End Time: 2025-04-26 19:11:47.524

Figure 6.9 Timestamp Log for Each Test

Table 6.1 Response Time Screen Scanning

Test Case Using	Response Time
Figure 6.5	7.216 seconds
Figure 6.6	5.478 seconds
Figure 6.7	5.132 seconds
Figure 6.8	7.444 seconds
Figure 6.9	7.679 seconds
Average Time	6.5898 seconds
Max Response Time	7.679 seconds
Min Response Time	5.132 seconds

Next, the audio scanning service, we will be using the same video shown in Figure 5.48 which is a scam voicemail recording. It is to be noted that the system flow in the audio scanning, will convert the audio to text and accumulate until there is enough words, then it will be sent to Gemini for analysis. Therefore, the response time depends on the scenario whether the person who is speaking has spoken sufficient number of words. However, to get a rough estimation of the service response time, the video used to test the service is assumed to have an average speed of speaking. The following is the result of the test.



```

1 //Audio Scanning Start Time Log
2 long startTime = System.currentTimeMillis();
3 Log.i("Tile-Listen", " Start time: " + startTime + " ms");
4
5 //Audio Scanning End Time Log
6 long endTime = System.currentTimeMillis();
7 Log.i("Tile-Listen", " End time: " + endTime + " ms");

```

Figure 6.10 Audio Scanning Timestamp Log Code

```

2025-04-26 21:25:39.712 5925-5925 Tile-Listen com.example.fyp_ver_2 I Start time: 1745673939712 ms
2025-04-26 21:25:49.661 5925-5925 Tile-Listen com.example.fyp_ver_2 I End time: 1745673949661 ms
2025-04-26 21:25:52.976 5925-5925 Tile-Listen com.example.fyp_ver_2 I End time: 1745673952976 ms
2025-04-26 21:25:57.150 5925-5925 Tile-Listen com.example.fyp_ver_2 I End time: 1745673957150 ms
2025-04-26 21:26:00.986 5925-5925 Tile-Listen com.example.fyp_ver_2 I End time: 1745673960986 ms

```

Figure 6.11 Audio Scanning Timestamp Log

In Figure 6.11 we can see there is a start time which indicates the service has started to listen to the surrounding audio. However, there is multiple end time, this end time indicates the Gemini has finished analysing the content and returned a result. Then it will wait for the next line of sentence and continue to analyse.

Table 6.2 Response Time Audio Scanning

Analysis Number	Time Taken for Result
1	9.949 seconds
2	3.315 seconds
3	4.174 seconds
4	3.836 seconds
Average	5.320 seconds
Max Time	9.950 seconds
Min Time	3.320 seconds

As for the secure browser, the response time will not be tested. This is due to the fact that the browser will depend on the type and the content that exists on the website. The more website links that are embedded on the website, the longer that it will take to scan and finally load the website. Therefore, if a response time test has been conducted on the service, it will not be meaningful as the time will significantly be different for each website.

6.1.3 Network Performance

At this section we will be testing the network performance. The application invokes two types of API services which are Gemini API and IPQS API. We will be integrating codes into the program to measure the time taken for each of the API to provide their response. Additionally, the test for Gemini API performance will be conducted using images from Figure 6.4 to Figure 6.9.



```

1 //Gemini Start Time Log
2 long startTime = System.currentTimeMillis();
3 String requestId = generateRequestId();
4 Log.i(TAG, "Gemini request " + requestId + " started at: " + startTime + " ms");
5
6 //Gemini End Time Log
7 long endTime = System.currentTimeMillis();
8 long executionTime = endTime - startTime;
9 Log.i(TAG, "Gemini request " + requestId + " completed at: " + endTime + " ms");
10 Log.i(TAG, "Gemini request " + requestId + " execution time: " + executionTime + " ms");

```

Figure 6.12 Gemini Timestamp Logging Code

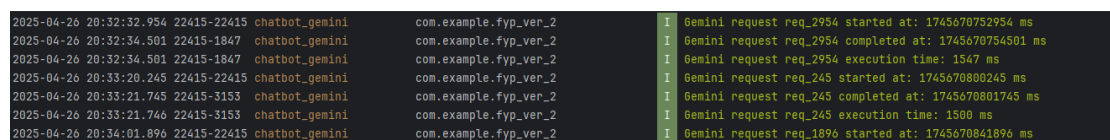


```

1 //IPQS Start Time Log
2 long startTime = System.currentTimeMillis();
3 String requestId = generateRequestId();
4 Log.i(TAG, "URL scan request " + requestId + " started at: " + startTime + " ms for URL: " + rawURL);
5
6 //IPQS End Time Log
7 long endTime = System.currentTimeMillis();
8 long totalExecutionTime = endTime - startTime;
9 Log.i(TAG, "URL scan request " + requestId + " total completion time: " + totalExecutionTime + " ms");

```

Figure 6.13 IPQS Timestamp Logging Code



2025-04-26 20:32:32.954	22415-22415	chatbot_gemini	com.example.fyp_ver_2	I Gemini request req_2954 started at: 1745670752954 ms
2025-04-26 20:32:34.501	22415-1847	chatbot_gemini	com.example.fyp_ver_2	I Gemini request req_2954 completed at: 1745670754501 ms
2025-04-26 20:32:34.501	22415-1847	chatbot_gemini	com.example.fyp_ver_2	I Gemini request req_2954 execution time: 1547 ms
2025-04-26 20:33:20.245	22415-22415	chatbot_gemini	com.example.fyp_ver_2	I Gemini request req_245 started at: 1745670800245 ms
2025-04-26 20:33:21.745	22415-3153	chatbot_gemini	com.example.fyp_ver_2	I Gemini request req_245 completed at: 1745670801745 ms
2025-04-26 20:33:21.746	22415-3153	chatbot_gemini	com.example.fyp_ver_2	I Gemini request req_245 execution time: 1500 ms
2025-04-26 20:34:01.896	22415-22415	chatbot_gemini	com.example.fyp_ver_2	I Gemini request req_1896 started at: 1745670841896 ms

Figure 6.14 Gemini Timestamp Log Part I

```
2025-04-26 20:34:03.158 22415-3772 chatbot_gemini com.example.fyp_ver_2 I Gemini request req_1896 completed at: 1745670843158 ms
2025-04-26 20:34:03.158 22415-3772 chatbot_gemini com.example.fyp_ver_2 I Gemini request req_1896 execution time: 1262 ms
2025-04-26 20:35:13.845 22415-22415 chatbot_gemini com.example.fyp_ver_2 I Gemini request req_3845 started at: 1745670913845 ms
2025-04-26 20:35:13.886 22415-22415 chatbot_gemini com.example.fyp_ver_2 I Gemini will be processing this context
```

Figure 6.15 Gemini Timestamp Log Part II

```
2025-04-26 20:35:15.360 22415-4086 chatbot_gemini com.example.fyp_ver_2 I Gemini request req_3845 completed at: 1745670915360 ms
2025-04-26 20:35:15.361 22415-4086 chatbot_gemini com.example.fyp_ver_2 I Gemini request req_3845 execution time: 1515 ms
2025-04-26 20:35:52.461 22415-22415 chatbot_gemini com.example.fyp_ver_2 I Gemini request req_2461 started at: 1745670952461 ms
2025-04-26 20:35:53.867 22415-4092 chatbot_gemini com.example.fyp_ver_2 I Gemini request req_2461 completed at: 1745670953867 ms
2025-04-26 20:35:53.867 22415-4092 chatbot_gemini com.example.fyp_ver_2 I Gemini request req_2461 execution time: 1486 ms
```

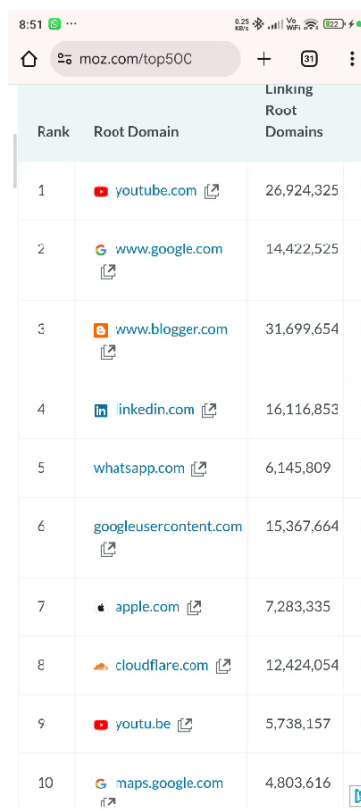
Figure 6.16 Gemini Timestamp Log Part III

Table 6.3 Network Performance Gemini API

Test Case Using	Response Time
Figure 6.5	1.547 seconds
Figure 6.6	1.500 seconds
Figure 6.7	1.262 seconds
Figure 6.8	1.151 seconds
Figure 6.9	1.406 seconds
Average Time	1.446 seconds
Max Response Time	1.547 seconds
Min Response Time	1.262 seconds

CHAPTER 6

Next, the IPQS API network performance. For this test we will be using 10 website links to benchmark the time it needed to return a result.



The screenshot shows a mobile browser interface with the address bar displaying 'moz.com/top500'. Below the address bar is a table with the following columns: Rank, Root Domain, and Linking Root Domains. The table lists the top 10 websites by ranking.

Rank	Root Domain	Linking Root Domains
1	youtube.com	26,924,325
2	www.google.com	14,422,525
3	www.blogger.com	31,699,654
4	linkedin.com	16,116,853
5	whatsapp.com	6,145,809
6	googleusercontent.com	15,367,664
7	apple.com	7,283,335
8	cloudflare.com	12,424,054
9	youtu.be	5,738,157
10	maps.google.com	4,803,616

Figure 6.17 IPQS Test Case

```
URL scan request url_4399 started at: 1745672174399 ms for URL: youtube.com
URL scan request url_4399 total completion time: 818 ms
URL scan request url_5229 started at: 1745672175229 ms for URL: www.google.com
URL scan request url_5229 total completion time: 534 ms
URL scan request url_5775 started at: 1745672175775 ms for URL: www.blogger.com
URL scan request url_5775 total completion time: 1633 ms
URL scan request url_7429 started at: 1745672177429 ms for URL: linkedin.com
URL scan request url_7429 total completion time: 1639 ms
URL scan request url_9084 started at: 1745672179084 ms for URL: whatsapp.com
URL scan request url_9084 total completion time: 626 ms
URL scan request url_9729 started at: 1745672179729 ms for URL: apple.com
URL scan request url_9729 total completion time: 470 ms
URL scan request url_220 started at: 1745672180220 ms for URL: cloudflare.com
URL scan request url_220 total completion time: 888 ms
URL scan request url_1127 started at: 1745672181127 ms for URL: youtu.be
URL scan request url_1127 total completion time: 608 ms
URL scan request url_1751 started at: 1745672181751 ms for URL: maps.google.com
URL scan request url_1751 total completion time: 779 ms
URL scan request url_2546 started at: 1745672182546 ms for URL: googleusercontent.com
URL scan request url_2546 total completion time: 445 ms
URL scan request url_3015 started at: 1745672183015 ms for URL: moz.com/top500
URL scan request url_3015 total completion time: 759 ms
```

Figure 6.18 IPQS Timestamp Log

Table 6.4 Network Performance IPQS API

Website Scanned	Response Time
youtube.com	818 ms
www.google.com	534 ms
www.blogger.com	1633 ms
linkedin.com	1639 ms
whatsapp.com	626 ms
apple.com	470 ms
cloudflare.com	888 ms
youtu.be	608 ms
maps.google.com	779 ms
googleusercontent.com	445 ms
Average Time	878ms
Max Response Time	1639 ms
Min Response Time	445 ms

A minor conclusion for this section is that all of the response time of the system services are tolerable and fast enough to ensure the user will not feel as if the application is unresponsive.

6.1.4 Scam Detection Accuracy with Gemini LLM

The benchmark method that was chosen to be used is the confusion matrix. The confusion matrix is a common method that is used to show the performance of a classification algorithm with a table [49]. Since, the result of our mobile application would be showing whether the content is safe or dangerous, it could be considered as a classification case. Although, after the system flags the content to be dangerous it will include the explanation of the LLM, but still, we are focusing on the fact did it flag the content properly at first before checking the explanation of the flag.

Classification is a supervised machine learning process of categorizing the input data into a variety of classes based on one or more features [50]. In our case, we are just using the confusion matrix that is used to benchmark classification cases. The confusion matrix consists of four basic elements which are used for calculation of accuracy, precision, recall, and F1 score. Those four basic elements are explained in a very simple manner according to the use case of our project in the following below:

1. **True Positive (TP)** : True positive means the system has flagged the content as dangerous and the actual content is dangerous. Correctly classified.
2. **True Negative (TN)** : True Negative means the system has flagged the content as dangerous, but the actual content is safe. Wrongly classified.
3. **False Positive (FP)** : False Positive means the system has flagged the content as safe and the actual content is safe. Correctly classified.
4. **False Negative (FN)** : False Negative means the system has flagged the content as safe, but the actual content is dangerous. Wrongly classified.

Using the four basic elements of the confusion matrix, we are able to compute the other four calculation result that are more informational. These four results are accuracy, precision, recall, and F1score. The following below would be explaining these four results:

1. **Accuracy:** Accuracy of the system is calculated by taking the correctly flagged content (TP+TN) divided by the total number of input samples (TP+FP+TN+FN).

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

2. **Precision:** Precision of the system is calculated by taking the correctly flagged dangerous content (TP) divided by the total number of sample contents that are predicted to be dangerous (TP+FP).

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

3. **Recall:** Recall of the system is calculated by taking the correctly flagged dangerous content (TP) divided by the total number of sample contents that has actual dangerous content (TP+FN)

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

4. **F1 score:** F1 score is calculated using the precision and recall value to give us an overall result of the model's performance.

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

The test will be using a public dataset with phishing emails to benchmark the Gemini model that we are using as our scam detector in our application. The dataset is found on Kaggle and promoted as phishing email detection dataset [51]. The dataset consists of 55950 number of emails where 11322 of the emails are safe emails and 7328 are phishing emails. However, for our use case we would just be testing 100 set of data to see the performance of the Gemini judgement against the phishing email without any tuning. Within the 100 emails, we have 62 safe emails and 38 phishing emails.

As a result :

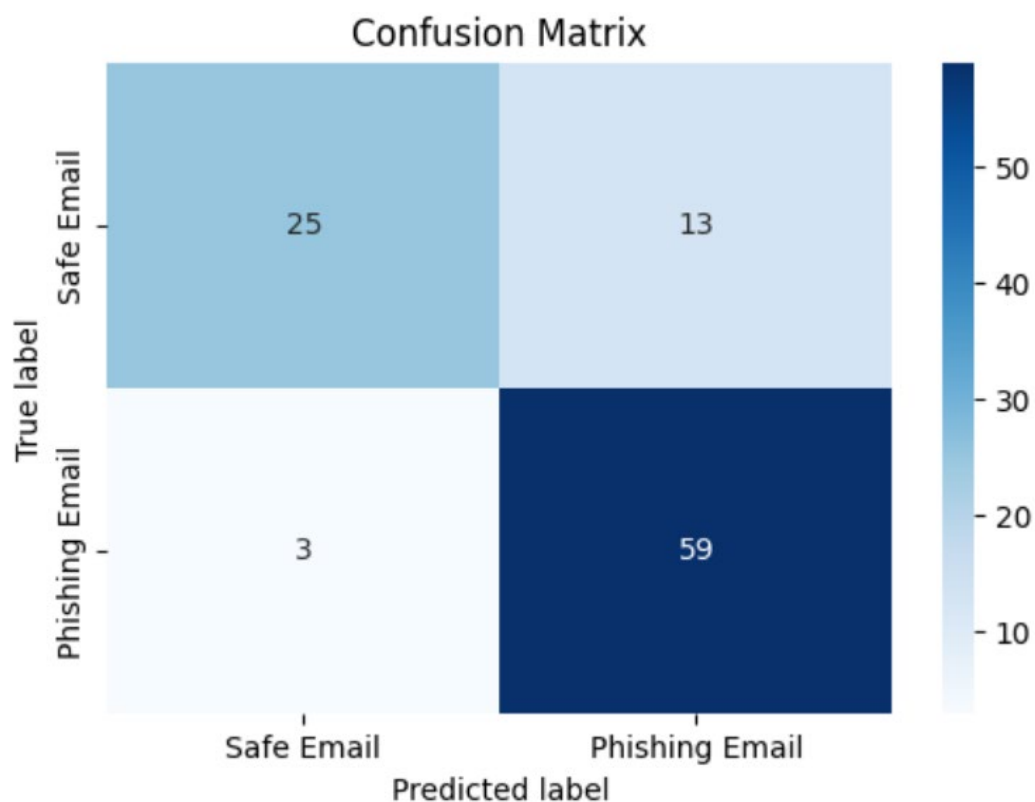


Figure 6.19 Confusion Matrix of Gemini with 100 Dataset

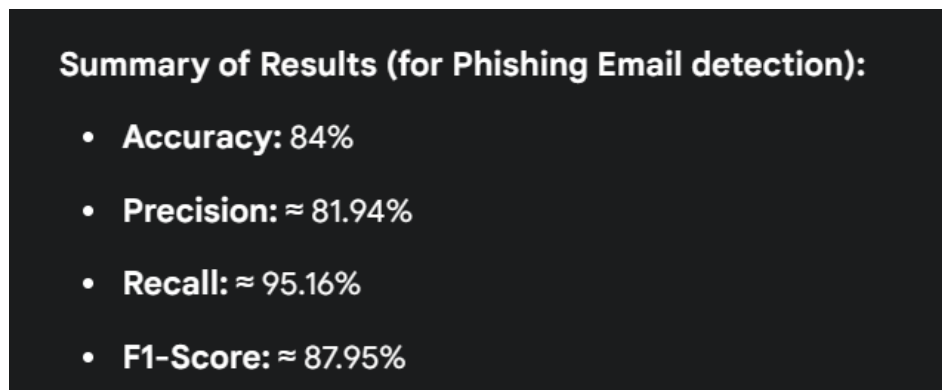


Figure 6.20 Number Scores of Gemini with 100 Dataset

As shown in Figure 6.20, we can see that the accuracy of the Gemini in detecting the phishing emails is quite high as it is above 80%. The testing has also calculated the precision, recall, and F1-Score to better visualize the performance of the model. Accuracy sometimes may not show enough to accurately conclude the performance.

The precision has scored 84% and recall has scored 84%, which is great and naturally the F1-Score is 87.95%. The precision score can be interpreted as how accurately can our model differentiate between safe and phishing email. Then the recall can be interpreted as how well can our model catch all phishing email. The F1-score is calculated with the precision and recall score, which it will show a value that is influenced by both value, if precision value is very low and recall value is high, the F1-score would be low also instead of getting influenced by the highest score.

In short, we can conclude that Gemini is a great candidate for our use case. However, in the future, the project development may look into using ChatGPT, a stronger LLM that is potentially more powerful than Gemini, to be used as our scam detector model or information validation model.

6.2 Testing Setup and Result

Within this section, we will utilize A/B testing methodology to determine the effectiveness of the solution developed. The A/B testing in simple is a way to compare two versions of something to figure out which performs better [52]. The users are given 4 rounds of test and total of 9 subjects to review. The following is the questionnaire that is distributed to the users for testing. The test consists of 5 users with each of them having 2 rounds. The first round will have the user to go through the test without the application assistance.

Task 1

As an average Malaysian citizen, you have received this message from JPJ stating your license will be deactivated soon.

Image title

What can you conclude *

Short answer text

Figure 6.21 Testing Task 1

Task 1 will require the user to spot the the Jabatan Pengangkutan Jalan (JPJ) message is scamming. Based on the user's response, all of the users whether with or without the developed application is able to flag that this message is a scam. This result is expected and provides us with assumption that the users have a certain level of awareness towards scams. Many of the test later on we will be able to observe that the question asked is "What can you conclude?". This is because if we were to place the question to "Whether is this malicious or not" it will hint the user more that the current task is

CHAPTER 6

malicious. However, this will affect the aim of the test, which is to try to distract the user to view the task as if they were normal, and based on their awareness to determine whether the messages that were given to them are malicious.

Then task 2,3,4 where the user is given with an Android smartphone to open up the Discord application. Discord is a voice, video, and text communication application [53]. The application is similar to the more common communication application WhatsApp.

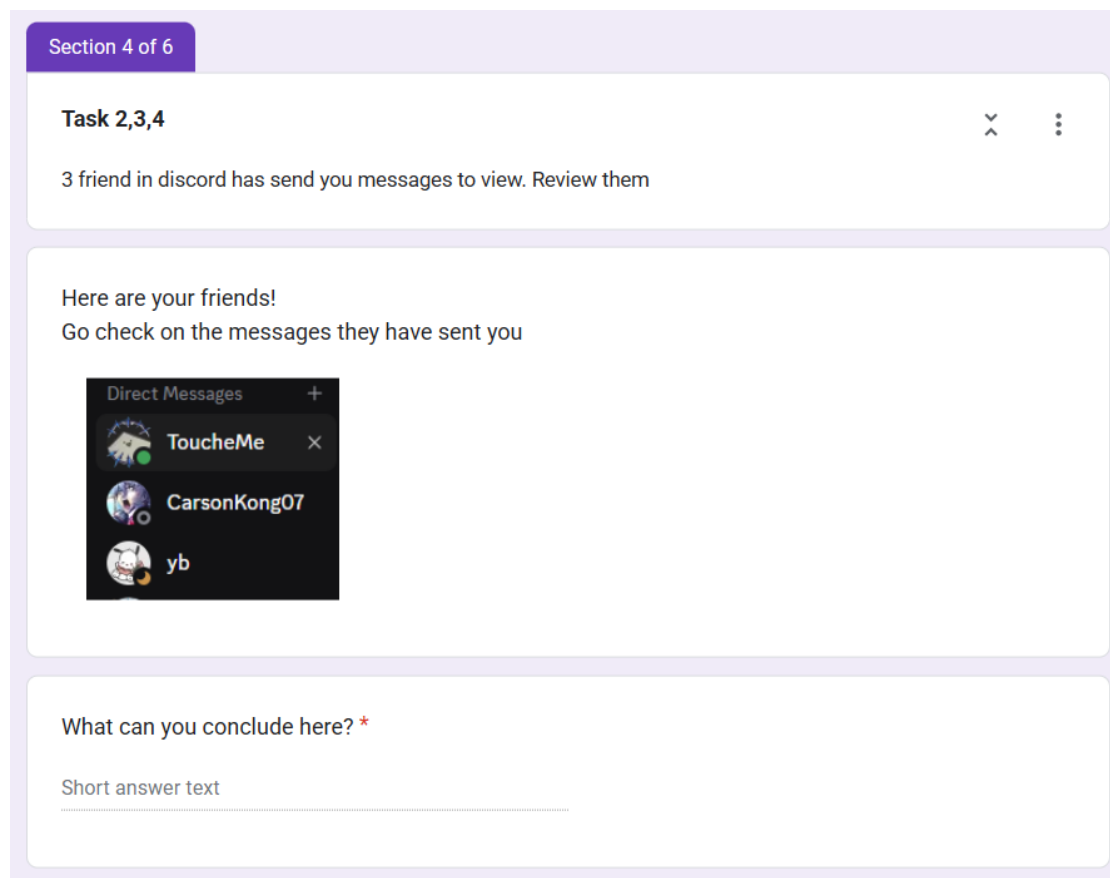


Figure 6.22 Testing Task 2, 3, 4

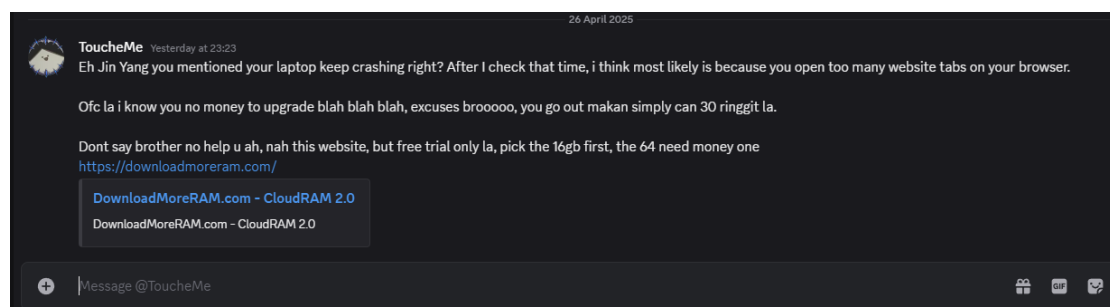


Figure 6.23 Message Received I

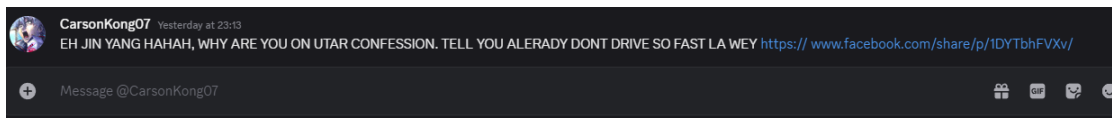


Figure 6.24 Message Received II

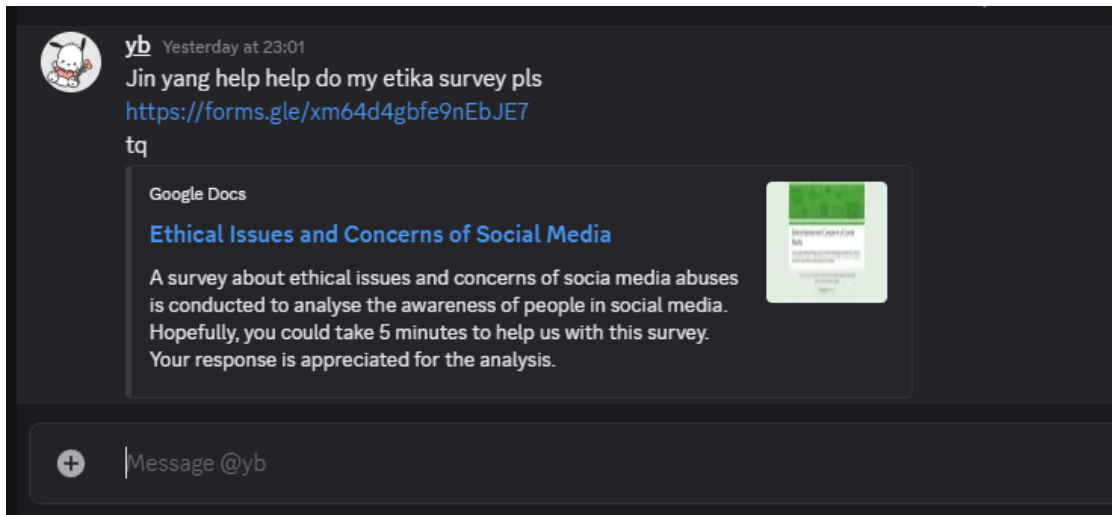


Figure 6.25 Message Received III

Each of my friend has send Discord messages to my Discord's account. The user will be using the Android smartphone login to my account to access these messages and review them. Later on, the users will be required to make a conclusion whether any of them are malicious, or this task is normal and there is no threat at all.

In context, the messages received, two of them are malicious. The "ToucheMe" account explained that the laptop kept crashing due to low memory on the laptop. Therefore, proposed an idea of downloading more memory on to the laptop. People with Information Technology (IT) background will more likely to spot that this is misleading or false information; however, other people may fall into this trap as they are not knowledgeable in this area.

CHAPTER 6

Then “CarsonKong07” has send a website link on surface is a Facebook post link. The website link that is on the surface is “[https:// www.facebook.com/share/p/1DYTbhFVXv/](https://www.facebook.com/share/p/1DYTbhFVXv/)”, but the actual website link that is being placed is “<https://fakebook.com/loginWillGetPranked>”. This method can be used by placing a text and a website link under this format in Discord:

[text](website.com)

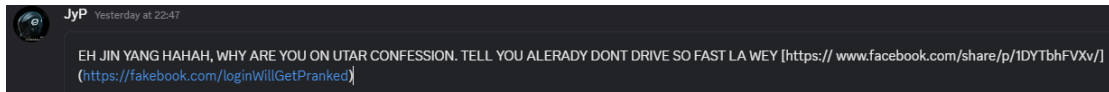


Figure 6.26 Building Decoy

The Facebook website link is a decoy and the actual link is Fakebook, a made up website link. If this were to be a real scenario, a threat actor may bring the user to malicious websites for them to login and capture their credentials.

Finally, the last message from “Yb” is a legit Google form.

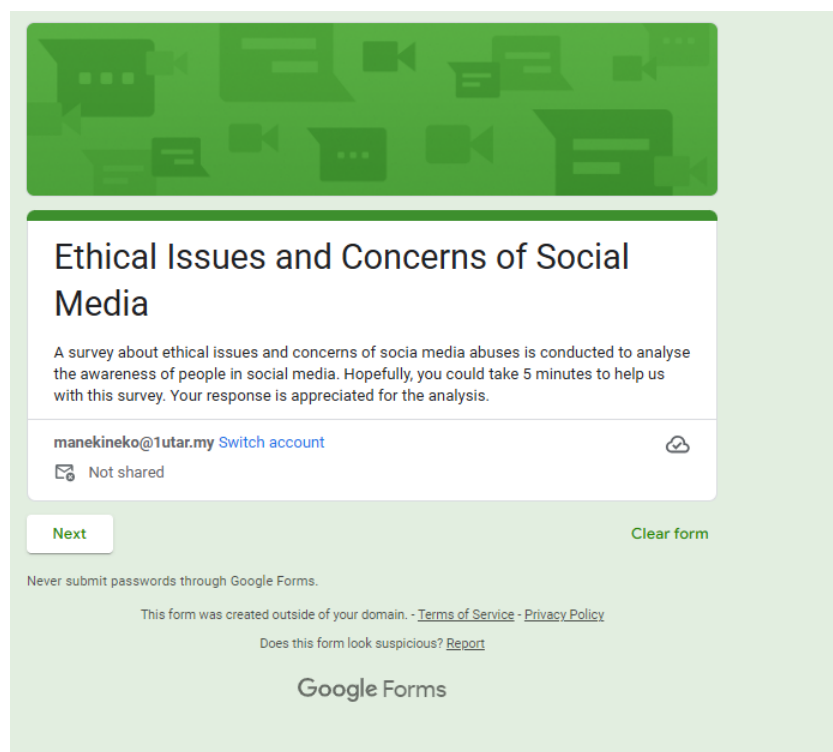
A screenshot of a Google Form titled "Ethical Issues and Concerns of Social Media". The form is displayed on a green background. The form content includes a title, a description, a user profile for "manekineko@tutar.my", and navigation buttons "Next" and "Clear form". The description text is: "A survey about ethical issues and concerns of socia media abuses is conducted to analyse the awareness of people in social media. Hopefully, you could take 5 minutes to help us with this survey. Your response is appreciated for the analysis." The user profile shows "manekineko@tutar.my" with a "Switch account" link and a "Not shared" status. The "Next" button is green and the "Clear form" button is green. At the bottom, there is a warning: "Never submit passwords through Google Forms." and a link to "Terms of Service - Privacy Policy". Below that, it says "This form was created outside of your domain." and a link to "Report". The Google Forms logo is at the bottom.

Figure 6.27 Received Message 3 Inside Link

The performance of the users without the application assistance can be seen below.

Email	Faculty	For this test were you given the application to	What can you conclude here?
hanyang7623@1utar.my	FICT	No	TouchMe unsure carsonkong07 and yb is fake link
yinghtoh@1utar.my	FICT	No	Nothing unusual
kyun0519q@1utar.my	FICT	No	The first and the third is real message, the second one is the scam message
tangje2005@gmail.com	Accounting	No	i will ignore second person
stepheniechai91@gmail	BF	No	No idea

Figure 6.28 Performance Task 2, 3, 4 without Application Assistance

We can see that 2 of the users reported that there is nothing unusual here. Then the rest of 3 user has suspicion that either 1 or 2 of the messages are malicious but none of them are able to fully spot all the malicious messages, some of them flagged the third message to be malicious while it was not. Then, the users are all going to the test again with the assistance of the application. It is to be noted that, they are not aware of the correctness of their answer on the first test run on this task.

Name	Email	Faculty	For this test were you given the application to	What can you conclude here?
TAN HAN YANG	hanyang7623@1utar.my	FICT	Yes	ToucheMe is a joke, CarsonKong07 is a fake and yo is safe link
Toh Hui Ying	yinghtoh@1utar.my	FICT	Yes	ToucheMe and yb message is relatively safe, Carson Kong is scamming me
KHOO YUN QIAN	kyun0519q@1utar.my	FICT	Yes	The first and second link is scam while the third is real message
Tang Jie En	tangje2005@gmail.com	Accounting	Yes	After use the app I will ignore the first person and second because if never use the app I
Chai Jia Yuan	stepheniechai91@gmail	BF	Yes	Could identify risky link , prevent users from being scam

Figure 6.29 Performance Task 2, 3, 4 with Application Assistance

Every user after the test with the application has commented either the fully correct answer or commented the application assisted to spot all the risky website links to prevent the users from being scam.

CHAPTER 6

Then the task 5,6,7 is regarding the cheapest deal for the JBL Boombox 3. The aim here is to distract the user from the real objective which is to ensure they are not scammed by the fake JBL website. The task can be seen at below:

Section 5 of 6

Task 5 ,6,7

Can you grab the cheapest deal you can for the JBL Boombox 3?

Who has the cheapest deal *

☐ JBL Website

☐ Lazada

☐ Shopee

Figure 6.30 Testing Task 5, 6, 7



Figure 6.31 Purchase Source 1 Fake JBL Website

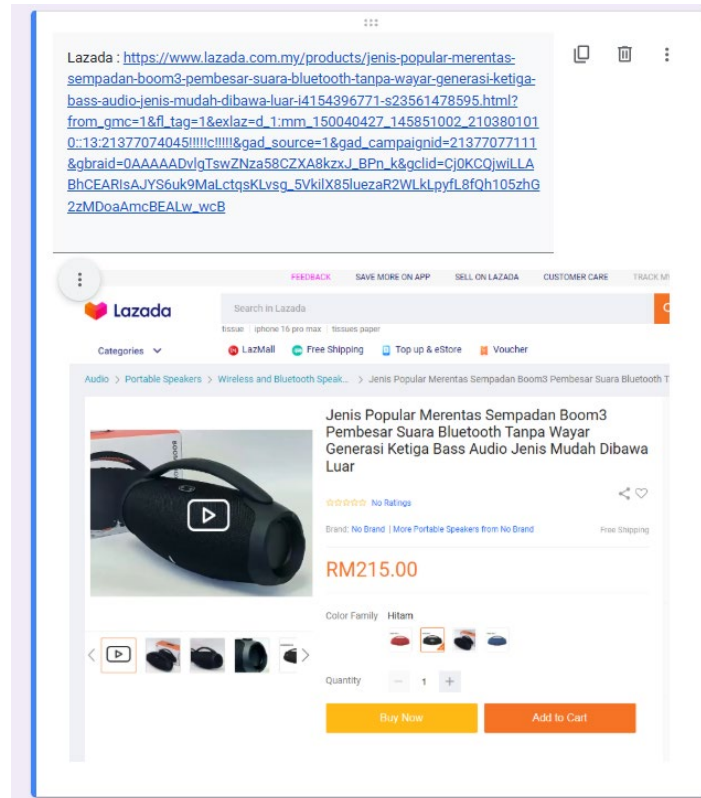


Figure 6.32 Purchase Source 2 Lazada Website

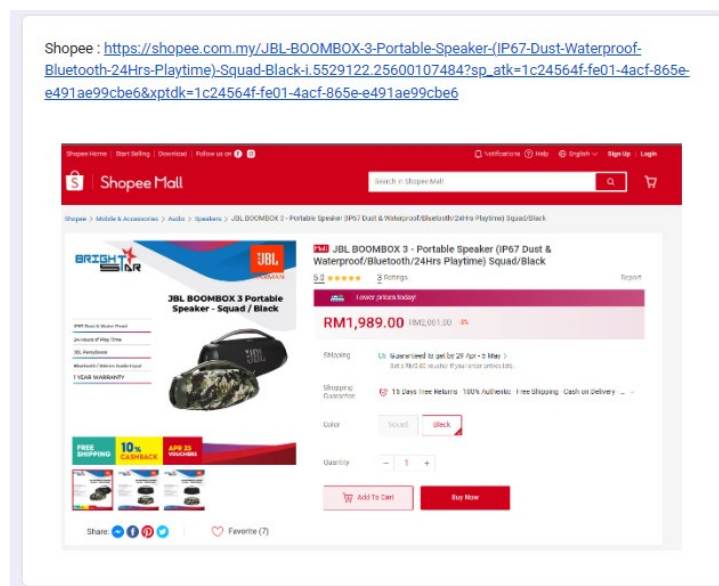


Figure 6.33 Purchase Source 3 Shopee Website

CHAPTER 6

The first option is the JBL website link that is provided by a famous China Press news Facebook account. The users does not know that this Facebook account is a replica of the official China Press Facebook account. It is distributing a highly similar JBL website to lure people purchase from it. As of now, April 26, 2025, the website still exists. Aside from phishing website it does not contain any malicious script.

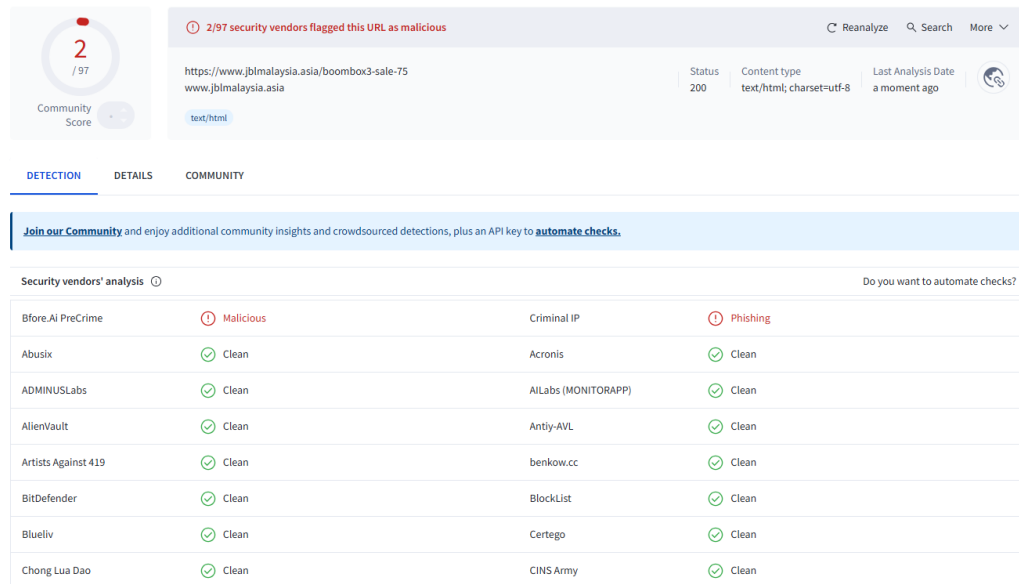


Figure 6.34 Virus Total Scan

CHAPTER 6

The second and third purchase link origin from the official Lazada and Shopee website. Both websites are e commerce platform in Asia [54] [55]. Although the Lazada website is a bit suspicious due to less amount of buyer on the product. Lazada does offer a feature “Buyer Confirmation” to hold the order amount and payment to the seller only once the user confirms [56]. All transaction is still under Lazada’s control, it is safer than the fake JBL website.

The user’s performance without the application assist is as shown below

Timestamp	Name	Email	Faculty	For this test were you given the application tc	Who has the cheapest deal
4/27/2025 15:44:46	TAN HAN YANG	hanyang7623@1utar.my	FICT	No	Lazada
4/27/2025 16:21:50	Toh Hui Ying	yinghtoh@1utar.my	FICT	No	JBL Website
4/27/2025 16:51:40	KHOO YUN QIAN	kyun0519q@1utar.my	FICT	No	Lazada
4/27/2025 19:20:26	Tang Jie En	tangje2005@gmail.com	Accounting	No	Lazada
4/27/2025 19:50:13	CHAI JIA YUAN	stepheniechai91@gmail	BF	No	JBL Website

Figure 6.35 Performance Task 5, 6, 7 without Application Assistance

With 2 users picking the JBL website as their purchasing option, and rest of the users picking the more common Lazada platform for the lower price. After the application is used on the same task

Timestamp	Name	Email	Faculty	For this test were you given the application tc	Who has the cheapest deal
4/27/2025 16:02:50	TAN HAN YANG	hanyang7623@1utar.my	FICT	Yes	Lazada
4/27/2025 16:30:20	Toh Hui Ying	yinghtoh@1utar.my	FICT	Yes	Lazada
4/27/2025 17:07:25	KHOO YUN QIAN	kyun0519q@1utar.my	FICT	Yes	Lazada
4/27/2025 19:34:47	Tang Jie En	tangje2005@gmail.com	Accounting	Yes	Lazada
4/27/2025 20:14:51	Chai Jia Yuan	stepheniechai91@gmail	BF	Yes	Lazada

Figure 6.36 Performance Task 5, 6, 7 with Application Assistance

All users have changed their decision to buy from the safer option, Lazada.

Task 8 and 9 is regarding two audio files that will be listened by the users can conclude what they are able to observe. The first audio is from the movie “The Wolf Of Wall Street – Selling Penny Stocks” where it tricks his client to purchase a risky stock and slowly manipulates his client to purchase those stocks. The main character is trying to misrepresent the actual company’s value and potential, create artificial demand via hype and pressure, and also profiting at the expense of unsuspecting investors. The transcript can be seen at the following.

“You mailed in my company a postcard a few weeks back requesting information on penny stocks that had huge upside potential with very little downside risk. Something just came across my desk, John. Name of the company, Aerotine International. It is a cutting-edge, high-tech firm out of the Midwest awaiting imminent patent approval on huge military and civilian applications. Now, right now, John.

The stock trades over the counter at 10 cents a share. Our analysts indicate it could go a heck of a lot higher than that. One thing I can promise you, even in this market, is that I never ask my clients to judge me on my winners. I ask them to judge me on my losers because I have so few. 4,000, that'd be 40,000 shares, John. Let me lock in that trade right now and get back to you with my secretary with an exact confirmation. Thank you for your vote of confidence, and welcome to the Investor Center.”

The second audio file is from a YouTube video made by Claude Diamond titled “How to Torture a Scam Telemarketer with G.U.T.S”. The audio file has been cut before the gentlemen exposes the call is a scam. This audio is to test whether the user listening to this is aware of the scenario. The conversation involves the caller explaining the donation can possibly be marked as a tax deductible fund. But in reality, the caller is trying to exploit the goodwill associated with the American veterans and pressuring the gentlemen (call receiver) to donate a potential significant portion. The transcript can be seen at the following.

Yeah, hi Joel, can you hear me? Hello? I was just saying, the American Veterans Support Foundation just kicked off their fundraising campaign and is a paid fundraiser for residential programs. We're sending out the new pledge kit to all supporting residents. Tragically, a lot of the same veterans that defended our country are now hungry, homeless, and cold. The goal of the drive is to help provide assistance with care packs for hospitalized veterans and food programs for homeless veterans. Now sir, we're asking for a one-time tax-deductible donation for the drive.

You said this is tax deductible, Joel? Joel? The American Veterans Support Foundation is designated by the IRS as a non-profit 501c3 charitable organization. Can I have your numbers? It may be tax deductible. However, we suggest you consult your tax preparer. Oh, OK. Can I have your 501c3 registration number then?

Users are accessing the audio via a laptop webpage of the Google drive that contains the audio file.

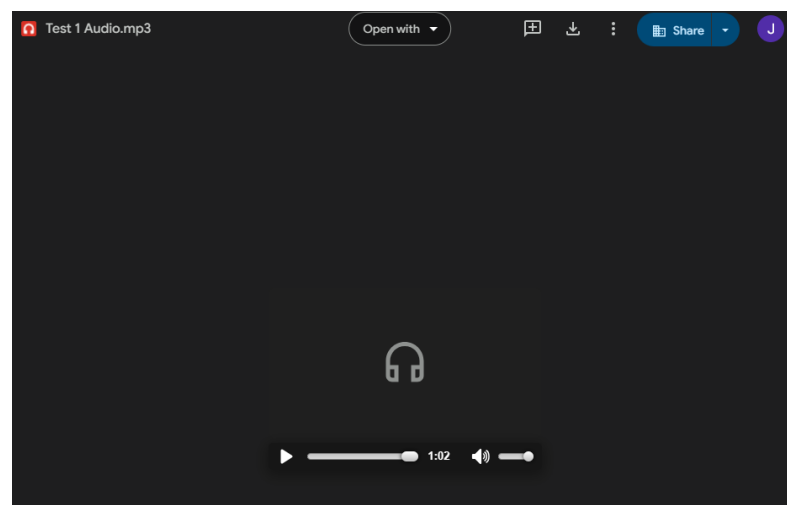


Figure 6.37 Audio File 1

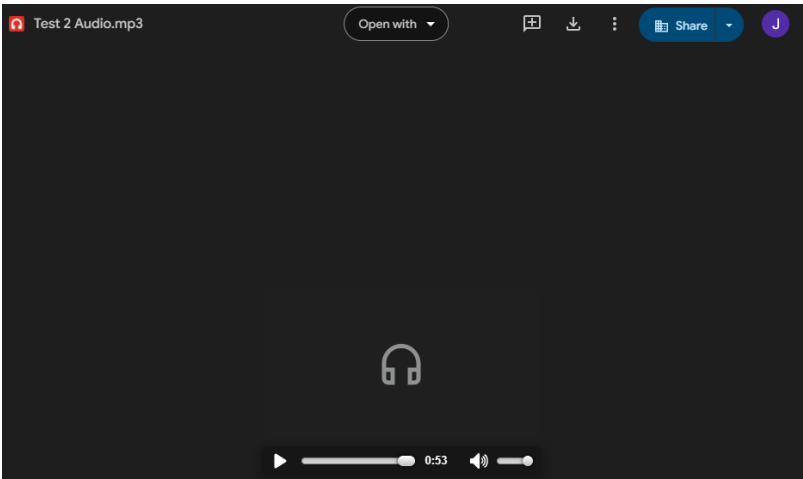


Figure 6.38 Audio File 2

The performance of the users without the application assistance can be seen below

			Audio 1 - What do you conclude	Audio 2 - What do you conclude
Email	Faculty	For this test were you given the application tc	https://drive.google.com/file/d/1ZvbfA2Dz/view?usp=sharing	https://drive.google.com/file/d/1ev478q7q01fexlU88Df6M89atMkGpe1cd/view?usp=sharing
hanyang7623@1utar.my	FICT	No	I don't understand	Fake call
yinghtoh@1utar.my	FICT	No	I don't know	It's a scam
kyun0519q@1utar.my	FICT	No	the audio speed was too fast causing not clear to	The audio 2 was blur than audio 1 , can't get the message state in the audio
tangje2005@gmail.com	Accounting	No	i don't understand	i don't know
stepheniechai91@gmail	BF	No	I could not catch up	Don't have a clear of listening environment

Figure 6.39 Performance Task 8, 9 without Application Assistance

In short, for the first audio, some users are not able to understand what is happening, and some users are not able to catch up with the conversation. Then for the second conversation, 2 users are able to flag this is a scam conversation and others commented that the audio was too hard to listen to.

CHAPTER 6

After the users has used the application to do the task, their performance has increased significantly.

Email	Faculty	For this test were you given the application to	Audio 1 - What do you conclude	Audio 2 - What do you conclude
hanyang7623@1utar.my	FICT	Yes	https://drive.google.com/file/d/1v2b0KzR1s1J0DmxK-CG01ZVt/view	https://drive.google.com/file/d/1w278a7a01fwU880tM0pMkGst1cdnew7uap-ahou/view
yinghtoh@1utar.my	FICT	Yes	Fake with confident	Fake with confident
kyun0519q@1utar.my	FICT	Yes	It's a scam	It's a scam
tangje2005@gmail.com	Accounting	Yes	A person is sharing a investment suggestion to his friend	This audio including scam message
stepheniechai91@gmail	BF	Yes	Ignore it	Ignore it
			I am more confident that this is a market scam.	I can suspect that this is a suspicious scam through donation with the assistant of this ai

Figure 6.40 Performance Task 8, 9 with Application Assistance

All of the users are able to correctly spot both scam or phishing conversation which confidence. The reason behind this is because the application assists the user in listening to the English conversation which is the first problem most user encounter, then secondly analyse the content of the conversation and understanding the context. Finally, providing the users with the analysis result. It is found common that many people are not able to listen to fluent and fast English conversation without subtitles. On top of that, the conversation has hidden their main goal which is to manipulate the end user. The application is emotionless and can keep capturing the conversation and constantly provides the users with the analysis. As a result, the users are equipped with a layer of protection when threat actors are trying to manipulate them.

6.3 Project Challenges

During the development of the project, one of the biggest issues that was faced is the lack of reference for similar project. For example, the coding and system design for the screen capture rarely has any example codes online for any similar project. As a result, the development of the application was based on trial and error. This has greatly increased the time needed for the application development.

Another project challenge is that the Audio listening feature originally was developed to intercept the phone call conversation between the user and the caller. However, the idea was not possible as the nature of the Android operating system does not allow two applications simultaneously use the audio system. For example, when the phone call application is active, then once the scam detection application is activated later on, the phone call application will be suspended and is not able to capture the user's mic or use the speaker to convey the caller's message.

Overall, the challenges are mostly lack of similar project with example codes. Unless with full knowledge of the entire Android operating system and development documentation, the development will be facing many trials and error.

6.4 Objectives Evaluation

The three objectives that were proposed for this project can be seen as below :

- a. To develop a privacy focused scam prevention and detection method to mitigate scam on smartphones
- b. To develop a scam message detection using LLM
- c. To develop scam audio detection using audio description and LLM

It is safe to say that this project has successfully achieved the three objective set initially. The final application developed offers screen scanning, audio scanning, and secure browse service. These three services are activated upon user's demand which offers user's privacy as it is not listening or capturing any data unless the user allows it to do so.

Based on Chapter 5 and Chapter 6, we are able to prove that the application can detect scams by leveraging LLM as the decision maker while additional services like IPQS to assist on scanning malicious URLs. The solution was proven to be effective as shown in Chapter 6 with the users' test result before and after the assistance of the developed application.

The final application prototype covers text scanning and audio scanning. Therefore, "To develop a scam message detection using LLM" and "To develop scam audio detection using audio description and LLM" has also been successfully achieved.

6.5 Concluding Remark

In short, this Chapter 6 has included the system self-testing, such as the load testing, response time, and network performance testing. Then, to test its effectiveness, the application has been utilized by 5 users with before and after performance review which has effectively improved the users' judgement on the test subjects.

CHAPTER 7

7.1 Conclusion

In conclusion, the initial proposed problem statements which are lack of vigilance and emotional triggers can be solved using the proposed plan. Both of the proposed problem statement is what had been concluded to be the primary factor that caused people falling victim into scams. The motivation of the project was that the current available technology specially the advance of LLM was capable of solving the problem but yet to be widely utilized to solve the problem. Then the objectives of this project are “to develop a privacy focused scam prevention and detection method to mitigate scam on smartphones”, “to develop a scam message detection using LLM”, and “to develop scam audio detection using audio description and LLM”. The proposed solution is developing a LLM powered anti-scam detector. The idea of the solution is that with minimized user interaction it could scan the user’s smartphone screen or listen to surrounding audio and alert the user when dangerous content is spotted.

Additionally, the objective “to develop scam audio detection using audio description and LLM” is achieved via capturing the surrounding sound instead of directly intercepting the phone call between the user and the caller. Therefore, at the current stage of the project it has provided the functionality to detect scam call, but the user would need to use the phone not to receive the call but to put close to the conversation and let the application to capture the audio.

Throughout the development, the project has shown to provide more than it had promised initially. The prototype mobile application could not only detect scams but also verify the information to prevent other dangerous intentions. Without a doubt all the services that are required to build the mobile application has been proven to be feasible using the current available services and technologies on the Internet. The success of this mobile application development was also made possible with the help of the large developer communities that have grown over the years for Android environments.

Originally the mobile application was purely designed to protect the user from scams by proposing the application to detect scam but the prototype mobile application that was built has been equipped with the capability of detecting not only scam information but also disinformation that could potentially harm the user. Additionally, the pop up feature of the

application was successfully built and enabled the user to instantly be alerted with the relevant information about the content that the user is currently viewing or listening.

The application that was proposed has met all the requirements and standards of the project; however, this would not be the end of this project as the application has much more potential.

It is hoped that this project would be in great use or reference to anyone that is working on similar topic. This project not only solves a real world problem but also has proposed a potentially marketable application to the market. Thank you

7.2 Recommendation/Future Work

The future direction of the project development is to build a more robust scam detection system logic and design. It is acknowledged that the current logic of scam detection is not perfect but is able to provide up to a level of effectiveness. In the future, the screen scanning feature may be mitigating recognizing the current screen image by sending the image to a image detection model to have another level of understanding of the current screen. This is done by recognition the User Interface (UI) elements. If done successfully, the screen scanning will be able to understand the current page is a web browser, communication application, photo gallery, etc.

Aside from that, the audio scanning feature has many improvements that can be done. One of which is to have a better system implementation of the audio input. The audio input of the current project is not able to capture words that are unclear, which still was a improvement over the first prototype where a certain degree of loudness was required for the application to capture the conversation.

Then finally, the secure browse. The secure browse is just an additional service that ensures if the user press on the website link before scanning it, it can be configured as a default browser and intercept visiting the website. It will be scanning the website before the user chooses to load it. This feature can be made to be more robust and separated from the application and offer more security layer.

References

- [1] MalaysiaKini, “Cybersecurity in Focus: Malaysia's Budget 2024 and the Fight Against Cyber Threats,” malaysiakini, 27 October 2023. [Online]. Available: <https://www.malaysiakini.com/announcement/684416>. [Accessed 3 March 2024].
- [2] A.-P. W. G. (APWG), “Phishing Activiyy Trends Report,” Anti-Phishing Working Group (APWG), 2 November 2023. [Online]. Available: https://docs.apwg.org/reports/apwg_trends_report_q1_2023.pdf. [Accessed 3 March 2024].
- [3] L. French, “OpenAI report reveals threat actors using ChatGPT in influence operations,” SC Media, 31 May 2024. [Online]. Available: <https://www.scmagazine.com/news/openai-report-reveals-threat-actors-using-chatgpt-in-influence-operations>. [Accessed 15 August 2024].
- [4] J. Shaw, “How the internet made it easier for all of us to be criminals, or victims,” Wired, 9 February 2019. [Online]. Available: <https://www.wired.com/story/julia-shaw-making-evil-internet-crime/>. [Accessed 15 August 2024].
- [5] H. Face, “GPT-J,” Hugging Face Corporation , [Online]. Available: https://huggingface.co/docs/transformers/en/model_doc/gptj. [Accessed 15 August 2024].
- [6] M. Elgan, “What to know about new generative AI tools for criminals,” Security Intelligence, 4 October 2023. [Online]. Available: <https://securityintelligence.com/articles/what-to-know-about-new-generative-ai-tools-for-criminals/>. [Accessed 15 August 2024].
- [7] S. Mahirova, “What is Worm GPT? The new AI behind the recent wave of cyberattacks,” Dazed, 18 July 2023. [Online]. Available: <https://www.dazeddigital.com/life-culture/article/60376/1/what-is-worm-gpt-the-new-ai-behind-the-recent-wave-of-cyberattacks>. [Accessed 15 August 2024].
- [8] G. P. Office, “Smartphone owners are now the global majority, New GSMA report reveals,” GSMA, 11 October 2023. [Online]. Available: [https://www.gsma.com/newsroom/press-release/smartphone-owners-are-now-the-global-majority-new-gsma-report-reveals/#:~:text=Smartphone%20owners%20are%20now%20the%20global%20majority%2C%20New%20GSMA%20report%20reveals,-Like%20what%20you&text=11th%20October%](https://www.gsma.com/newsroom/press-release/smartphone-owners-are-now-the-global-majority-new-gsma-report-reveals/#:~:text=Smartphone%20owners%20are%20now%20the%20global%20majority%2C%20New%20GSMA%20report%20reveals,-Like%20what%20you&text=11th%20October%20). [Accessed 15 August 2024].

Appendix A : WEEKLY REPORT

- [9] C. Insurance, “Mobile Phones Likely to Play Larger Role in Cyber Attacks, Experts Predict,” Chivarolie & Associates Insurance Services, 18 January 2022. [Online]. Available: <https://chivaroli.com/mobile-phones-likely-to-play-larger-role-in-cyber-attacks-experts-predict/>. [Accessed 15 August 2024].
- [10] P. Dizikes, “Study: On Twitter, false news travels faster than true stories,” MIT News , 8 March 2018. [Online]. Available: <https://news.mit.edu/2018/study-twitter-false-news-travels-faster-true-stories-0308>. [Accessed 15 August 2024].
- [11] D. B. Hasnain, “Are humans emotional or rational?,” Tribune, 5 June 2023. [Online]. Available: <https://tribune.com.pk/story/2420247/are-humans-emotional-or-rational>. [Accessed 15 August 2024].
- [12] A. Fatrah, “Improve the quality of your OCR information extraction,” Medium, 20 March 2022. [Online]. Available: <https://aicha-fatrah.medium.com/improve-the-quality-of-your-ocr-information-extraction-ebc93d905ac4#:~:text=Tesseract%20works%20best%20on%20images,rapidly%20below%208pt%20x%20300dpi..> [Accessed 25 August 2024].
- [13] A. Fatrah, “Improve the quality of your OCR information extraction,” Medium, 20 March 2022. [Online]. Available: <https://aicha-fatrah.medium.com/improve-the-quality-of-your-ocr-information-extraction-ebc93d905ac4#:~:text=Tesseract%20works%20best%20on%20images,rapidly%20below%208pt%20x%20300dpi..> [Accessed 19 August 2024].
- [14] Falade and P. Victor, “Trend and Emerging Types of “419” SCAMS. Proceedings of the Cyber Secure Nigeria Conference,” arXiv, 23 August 2023. [Online]. Available: <https://doi.org/10.48550/arXiv.2308.12448>. [Accessed 28 March 2024].
- [15] P. A. Department, “Annual Scams and Cybercrime Brief 2022,” Singapore Police Force, 8 February 2023. [Online]. Available: <https://www.police.gov.sg/-/media/8F06592D8FBE475C8D2B92EB3BFFE7FC.ashx#:~:text=The%20number%20of%20scam%20and,from%2031%2C728%20cases%20in%202022..> [Accessed 28 March 2024].
- [16] A. Derakhshan, I. G. Harris and M. Behzadi, “Detecting Telephone-based Social Engineering Attacks using Scam Signatures,” ACM Digital Library, 26 April 2021. [Online]. Available: <https://dl.acm.org/doi/pdf/10.1145/3445970.3451152>. [Accessed 28 March 2024].

Appendix A : WEEKLY REPORT

- [17] S. S. Roy, P. Thota, K. V. Naragam and S. Nilizadeh, “From Chatbots to PhishBots? - Preventing Phishing scams created using ChatGPT,,” The University of Texas at Arlington, Arlington, Texas, 2024.
- [18] H. Mahari, “Online fraud cases in M'sia doubled over 5 years, CCID director warns of escalating threat,” New Straits Times, 3 March 2024. [Online]. Available: <https://www.nst.com.my/news/crime-courts/2024/03/1020542/online-fraud-cases-msia-doubled-over-5-years-ccid-director-warns#:~:text=Ramli%20said%20that%20the%20portal,be%20involved%20in%20fraudulent%20activities..> [Accessed 29 March 2024].
- [19] C. C. I. D. R. M. Police, “Semakmule Portal,” Commercial Crime Investigation Department Royal Malaysia Police, 2020. [Online]. Available: <https://semakmule.rmp.gov.my/>. [Accessed 29 March 2024].
- [20] D. Cox, “Truecaller Review: 5 Things To Know Before Downloading the App,” Clark, 26 September 2022. [Online]. Available: <https://clark.com/cell-phones/truecaller-review/>. [Accessed 29 March 2024].
- [21] Truecaller, “About Us,” Truecaller, [Online]. Available: <https://www.truecaller.com/about>. [Accessed 28 March 2024].
- [22] A. Frid, “Truecaller Unveils A New Brand Identity and Upgraded AI Identity Features for Fraud Prevention,” Nasdaq , 18 September 2023. [Online]. Available: <https://attachment.news.eu.nasdaq.com/a898bf32598cdaded068f358b99dd7f9e>. [Accessed 29 March 2024].
- [23] whoscall, “whoscall,” whoscall, [Online]. Available: <https://whoscall.com/en>. [Accessed 29 March 2024].
- [24] Gogolook, “Caller Database Solutions,” Gogolook, [Online]. Available: <https://gogolook.com/en/caller-database-solutions>. [Accessed 29 March 2024].
- [25] ScamAdviser, “About ScamAdviser,” ScamAdviser, [Online]. Available: <https://www.scamadviser.com/about-scamadviser>. [Accessed 18 August 2024].
- [26] M. Tladi, “Is ScamAdviser Legit? Here's What You Need to Know,” Make Use Of, 16 August 2024. [Online]. Available: <https://www.makeuseof.com/is-scamadviser-legit-heres-what-you-need-to->

Appendix A : WEEKLY REPORT

know/#:~:text=ScamAdviser%20is%20a%20tool%20that,on%20a%20range%20of%20factors.
[Accessed 18 August 2024].

- [27] Domain.com, “What is WHOIS and How Is It Used?,” Domain.com, Natalie Brownell. [Online]. Available: <https://www.domain.com/blog/what-is-whois-and-how-is-it-used/>. [Accessed 18 August 2024].
- [28] Trustpilot, “Truecaller review,” Trustpilot, [Online]. Available: <https://www.trustpilot.com/review/www.truecaller.com>. [Accessed 17 April 2024].
- [29] Apple, “App Store Preview Truecaller: Spam Call Blocker,” Apple, [Online]. Available: <https://apps.apple.com/us/app/truecaller-spam-call-blocker/id448142450>. [Accessed 17 April 2024].
- [30] Apple, “App Store Preview Whoscall - Caller ID & Block,” Apple, [Online]. Available: <https://apps.apple.com/my/app/whoscall-caller-id-block/id929968679>. [Accessed 17 April 2024].
- [31] malaymail, “Bukit Aman CCID director tells public 160,000 bank accounts, phone numbers related to scammers listed in SemakMule database,” malaymail, 4 January 2024. [Online]. Available: <https://www.malaymail.com/news/malaysia/2024/01/04/bukit-aman-ccid-director-tells-public-160000-bank-accounts-phone-numbers-related-to-scammers-listed-in-semakmule-database/110652>. [Accessed 29 March 2024].
- [32] Google, “Explore vision capabilities with the Gemini API,” Google, 5 August 2024. [Online]. Available: <https://ai.google.dev/gemini-api/docs/vision?lang=python>. [Accessed 25 August 2024].
- [33] egorpugin, “tesseract-ocr/tesseract,” Github, 23 August 2024. [Online]. Available: <https://github.com/tesseract-ocr/tesseract>. [Accessed 25 August 2024].
- [34] Amazon, “What is OCR (Optical Character Recognition)?,” Amazon, [Online]. Available: <https://aws.amazon.com/what-is/ocr/#:~:text=A%20simple%20OCR%20engine%20works,is%20called%20optical%20word%20recognition..> [Accessed 3 September 2024].
- [35] R. Smith, “An Overview of the Tesseract OCR Engine,” Google, [Online]. Available: <https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/33418.pdf>. [Accessed 3 September 2024].

- [36] MHarris1319, “Tesseract OCR won't recognize a letter I in sans-serif as a letter I, even if pipe and exclamation are blacklisted,” StackOverFlow, 2022. [Online]. Available: <https://stackoverflow.com/questions/73439267/tesseract-ocr-wont-recognize-a-letter-i-in-sans-serif-as-a-letter-i-even-if-pi>. [Accessed 25 August 2024].
- [37] Google, “Recognize text in images with ML Kit on Android,” Google, 23 August 2024. [Online]. Available: <https://developers.google.com/ml-kit/vision/text-recognition/v2/android>. [Accessed 25 August 2024].
- [38] Google, “ML Kit Guides,” Google, 23 August 2024. [Online]. Available: <https://developers.google.com/ml-kit/guides>. [Accessed 3 September 2024].
- [39] Google, “Text Recognition v2 Guide,” Google, 10 July 2024. [Online]. Available: <https://developers.google.com/ml-kit/vision/text-recognition/v2>. [Accessed 3 September 2024].
- [40] Google, “Introducing Gemini: our largest and most capable AI model,” Google, 6 December 2023. [Online]. Available: <https://blog.google/technology/ai/google-gemini-ai/#sundar-note>. [Accessed 28 April 2025].
- [41] T. Team, “Transformer Architecture: Redefining Machine Learning Across NLP and Beyond,” Toloka, 6 July 2024. [Online]. Available: <https://toloka.ai/blog/transformer-architecture/>. [Accessed 28 April 2025].
- [42] G. F. Geeks, “Self-Supervised Learning (SSL),” Geeks For Geeks, 13 December 2023. [Online]. Available: <https://www.geeksforgeeks.org/self-supervised-learning-ssl/>. [Accessed 28 April 2025].
- [43] N. S. N. P. J. U. L. J. A. N. G. L. K. I. P. Ashish Vaswani, “Attention Is All You Need,” Cornell University, 12 June 2017. [Online]. Available: <https://arxiv.org/abs/1706.03762>. [Accessed 28 April 2025].
- [44] Amanatullah, “Transformer Architecture explained,” Medium, 1 September 2023. [Online]. Available: <https://medium.com/@amanatulla1606/transformer-architecture-explained-2c49e2257b4c>. [Accessed 28 April 2025].
- [45] I. Q. Score, “Malicious URL Scanner API Documentation,” IP Quality Score, [Online]. Available: <https://www.ipqualityscore.com/documentation/malicious-url-scanner-api/overview>. [Accessed 4 September 2024].

- [46] Google, “Recognition Listener,” Google, 11 April 2024. [Online]. Available: <https://developer.android.com/reference/android/speech/RecognitionListener>. [Accessed 3 September 2024].
- [47] D. Shah, “What is In-context Learning, and how does it work: The Beginner’s Guide,” Lakera, 26 March 2025. [Online]. Available: <https://www.lakera.ai/blog/what-is-in-context-learning>. [Accessed 21 April 2025].
- [48] A. Developers, “Profile your app performance,” Google, 29 October 2024. [Online]. Available: <https://developer.android.com/studio/profile>. [Accessed 25 April 2025].
- [49] P. Singh and K. K. S. A. S. Narendra Singh, “Machine Learning and the Internet of Medical Things in Healthcare,” ScienceDirect, 2021. [Online]. Available: <https://www.sciencedirect.com/topics/engineering/confusion-matrix#chapters-articles>. [Accessed 19 August 2024].
- [50] M. Ramakrishnan, “Home / Blog / Artificial Intelligence and Machine Learning,” Emeritus, 24 July 2023. [Online]. Available: <https://emeritus.org/blog/artificial-intelligence-and-machine-learning-classification-in-machine-learning/#:~:text=In%20machine%20learning%2C%20classification%20is,datasets%20of%20input%20and%20output..> [Accessed 22 August 2024].
- [51] C. Cop, “Phishing Email Detection,” Kaggle, 2023. [Online]. Available: <https://www.kaggle.com/datasets/subhajournal/phishingemails?resource=download>. [Accessed 25 August 2024].
- [52] A. Gallo, “A Refresher on A/B Testing,” Harvard Business Review, 28 June 2017. [Online]. Available: <https://hbr.org/2017/06/a-refresher-on-ab-testing>. [Accessed 23 April 2025].
- [53] Discord, “What is Discord?,” Discord, 12 May 2022. [Online]. Available: <https://discord.com/safety/360044149331-what-is-discord>. [Accessed 12 April 2025].
- [54] Lazada, “About Us,” Lazada, [Online]. Available: <https://www.lazada.com/en/about/>. [Accessed 27 April 2025].
- [55] Shopee, “About Us,” Shopee, [Online]. Available: <https://careers.shopee.com.my/about>. [Accessed 27 April 2025].

- [56] M. o. H. Affairs, “Tips for transacting safely on Lazada,” Ministry of Home Affairs , [Online]. Available: <https://www.mha.gov.sg/e-commerce-marketplace-transaction-safety-ratings/lazada>. [Accessed 27 April 2025].
- [57] B. D. James, P. A. Boyle and D. A. Bennett, “Correlates of Susceptibility to Scams in Older Adults Without Dementia,” National Library of Medicine, 1 January 2015. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3916958/>. [Accessed 17 August 2024].
- [58] J. L. M. Stacey J. Drubner, “Teens and Young Adults are at High-risk for Online Scams,” Mass General Brigham, 1 January 2024. [Online]. Available: https://eap.partners.org/news_posts/young-people-at-high-risk-for-online-scams/#:~:text=Research%20indicates%20that%20older%20adults,of%20losing%20control%20of%20devices.. [Accessed 17 August 2024].
- [59] Norton, “Your free AI-powered scam detector,” Norton, 2024. [Online]. Available: <https://malaysia.norton.com/products/genie-scam-detector>. [Accessed 18 August 2024].
- [60] N. J. Rubenking, “Norton Genie Review,” PC MAG, 28 July 2023. [Online]. Available: <https://www.pcmag.com/reviews/norton-genie>. [Accessed 18 August 2024].
- [61] Phoneia, “Android 10 allows two apps different access to the microphone at the same time, to the delight of Assistant,” Phoneia, 16 September 2019. [Online]. Available: <https://phoneia.com/en/android-10-allows-two-apps-different-access-to-the-microphone-at-the-same-time-to-the-delight-of-assistant/>. [Accessed 16 August 2024].
- [62] tesseract-ocr, “Tesseract User Manual,” tesseract-ocr, [Online]. Available: <https://tesseract-ocr.github.io/tessdoc/>. [Accessed 19 August 2024].
- [63] OpenAI, “Introducing GPTs,” OpenAI, 6 November 2023. [Online]. Available: <https://openai.com/blog/introducing-gpts>. [Accessed 3 March 2024].
- [64] Amazon, “What is an Audio-To-Text Converter?,” Amazon, [Online]. Available: <https://aws.amazon.com/what-is/audio-to-text-converter/#:~:text=Acoustic%20speech%20recognition%20technology%20matches,to%20text%20in%20many%20languages..> [Accessed 4 September 2024].
- [65] J. Manyika and S. Hsiao, “An overview of the Gemini app,” Google, [Online]. Available: <https://gemini.google/overview-gemini->

Appendix A : WEEKLY REPORT

app.pdf#:~:text=Gemini%20is%20powered%20by%20Google's,rules%20and%20model%2Dbased%20classifiers.. [Accessed 4 September 2024].

POSTER

