

In the name of ALLAH, the Most Benevolent, the Most Courteous

# QR SHIELD



By

**Capt Muhammad Bilal Mehmood**

**Capt Sheikh Hamza Aftab**

**GC Suleman Munawar**

Supervised by:

**Major Bilal Ahmed**

Submitted to the faculty of Department of Information Security  
Military College of Signals, National University of Sciences and Technology, Islamabad,  
in partial fulfillment for the requirements of B.E Degree in Information Security.

June 2024

## **CERTIFICATE OF CORRECTNESS AND APPROVAL**

*This is to officially state that the thesis work contained in this report*

**“QR SHIELD”**

*is carried out by*

**Capt Muhammad Bilal Mehmood**

**Capt Sheikh Hamza Aftab**

**&**

**GC Suleman Munawar**

*under my supervision and that in my judgement, it is fully ample, in scope and excellence, for the*

*degree of Bachelor of Information Security*

*in Military College of Signals, National University of Sciences and Technology (NUST),*

*Islamabad.*

**Approved by**

\_\_\_\_\_  
**Supervisor**  
**MAJOR BILAL AHMED**  
**Department of IS, MCS**

Date: \_\_\_\_\_

## **DECLARATION OF ORIGINALITY**

We hereby declare that no portion of work presented in this thesis has been submitted in support of another award or qualification in either this institute or anywhere else.

## **ACKNOWLEDGEMENTS**

Allah Subhan'Wa'Tala is the sole guidance in all domains.

Our parents, colleagues and most of all supervisor, Major Bilal Ahmed without your guidance.

The group members, who through all adversities worked steadfastly.

## **Plagiarism Certificate (Turnitin Report)**

This thesis has \_\_\_\_ similarity index. Turnitin report endorsed by Supervisor is attached.

---

Capt Muhammad Bilal Mehmood

00000359423

---

Capt Sheikh Hamza Aftab

00000359389

---

00000358982

GC Suleman Munawar

---

Supervisor M

Major Bilal Ahmed

## **ABSTRACT**

QR Shield is an innovative Android application designed to fortify cybersecurity defenses against QR code-based cyber threats, ultimately fostering a safer and more vigilant society. At its core,

QR Shield empowers users with a multi-layered defence mechanism: QR Code Scanning: The application provides users with the option to either scan QR codes directly or select them from the gallery, ensuring versatility and convenience in QR code handling. Content Extraction and Analysis: Upon scanning, QR Shield extracts the content embedded within the QR code. This content is then swiftly transmitted to VirusTotal via its API for comprehensive scanning and analysis. VirusTotal Integration: Leveraging the robust capabilities of VirusTotal, QR Shield conducts thorough scans to detect any malicious elements present within the QR code content. This includes malware, phishing attempts, and other forms of cyber threats. Analysis and Reporting: The application meticulously examines the scan results, providing users with valuable insights. This includes the number of antivirus engines flagging the content as malicious, along with detailed information on the nature of the detected threats. Risk Awareness and Decision Support: Armed with comprehensive scan reports, QR Shield educates users on the potential risks associated with the scanned content. It prompts users to make informed decisions by cautioning them about the potential threat and allowing them to proceed or abort the action. Enhanced Cybersecurity: QR Shield acts as a proactive barrier against QR code-based cyber attacks and frauds, safeguarding users' devices and personal information from potential threats lurking within innocent-looking QR codes. Heightened Awareness: By providing detailed insights into the nature of detected threats, QR Shield raises awareness among users about the diverse forms of cyber threats prevalent in the digital landscape, fostering a more cyber-literate society. Empowerment Through Information: Through its transparent reporting and decision support mechanisms, QR Shield empowers users to make informed choices regarding their online interactions, empowering them to navigate the digital realm with confidence and caution. Prevention of Cyber Crimes: By pre-emptively scanning QR codes for malicious content, QR Shield helps thwart cyber crimes such

as identity theft, financial fraud, and malware dissemination, contributing to a safer and more secure online environment for all. In essence, QR Shield transcends the traditional boundaries of QR code scanning applications by not only detecting malicious content but also serving as a beacon of cyber awareness and empowerment, thereby championing the cause of cybersecurity and societal well-being.

## Table of Contents

<b>Chapter 1: Introduction.....</b>	<b>1</b>
1.1 Overview.....	4
1.2 Problem Statement.....	4
1.3 Proposed Solution.....	7
1.4 Working Principle.....	8
1.5 Objectives.....	9
1.5.1 General Objectives: .....	9
1.5.2 Academic Objectives: .....	9
1.6 Scope.....	9
1.7 Deliverables.....	10
1.7.2 QR Shield Application.....	10
1.7.3 Technical Documentation.....	10
1.7.4 User Manual.....	10
1.7.4 Source Code.....	10
1.8 Relevant Sustainable Development Goals .....	11
1.9 Structure of Thesis.....	11
<b>Chapter 2: Literature Review.....</b>	<b>12</b>
2.1 Industrial background of QR Code Technology.....	13
2.2 Existing Solutions and Their Drawbacks.....	16
2.3 Overview of Existing QR Code Scanning Applications.....	16
2.4 Identification and discussion of limitations.....	17
2.5 Examples of Common Drawbacks.....	18
2.6 Review of Relevant Studies and Research Papers.....	18
2.7 Discussion on Implications for User Security and Privacy.....	19
2.8 Research Papers on QR Code Security.....	19
2.9 Overview of Innovative Approaches and Methodologies.....	20
<b>Chapter 3: Design and Development.....</b>	<b>22</b>
3.1 Design Process.....	22
3.1.1 User Research.....	22
3.1.2 Information Architecture.....	23
3.1.3 Flowchart Diagram.....	24
3.1.4 Wireframing.....	25
3.1.5 Prototyping.....	25
3.1.6 Visual Design.....	25
3.1.7 High-Fidelity Prototyping.....	25
3.1.8 Usability Testing.....	25
3.2 User Interface Features.....	25
3.2.1 QR Code Scanning Display.....	25
3.2.2 Scan History Display.....	27
3.2.3 Alert Notifications Display Screen.....	28
3.3 Development Process.....	29
3.3.1 Requirement Analysis.....	29
3.3.2 Development Phase.....	30
3.3.2.1 Frontend Development.....	30
3.3.2.2 Backend Development.....	30



3.3.2.3 Integration with Virus Total API.....	30
3.3.3 Testing Phase.....	31
3.3.3.1 Unit Testing .....	31
3.3.3.2 Functional Testing .....	31
3.3.4 Deployment Phase.....	33
<b>Chapter 4: Code Analysis and Evaluation.....</b>	<b>34</b>
4.1 ScanningActivity Code Analysis.....	34
4.1.1 Imports.....	34
4.1.2 Class Members .....	34
4.1.3 onCreate() Method .....	35
4.1.4 openPhotoPicker() Method .....	35
4.1.5 startCamera() Method .....	35
4.1.6 analyzeImage() Method .....	36
4.1.7 analyzeUrl() and getReport() Methods.....	37
4.1.8 Permissions Handling.....	38
4.2 Evaluation of Scanning Activity Code .....	38
4.2.1 API Key Exposure.....	38
4.3 Result Activity Code Analysis.....	39
4.3.1 Imports.....	39
4.4 Evaluation of Result Activity Code.....	41
4.4.1 Null Safety.....	41
<b>Chapter 5: Conclusion.....</b>	<b>42</b>
<b>Chapter 6: Future Work.....</b>	<b>43</b>
<b>References and Work Cited.....</b>	<b>50</b>

## List of Figures

Figure 1: QR Shield Main Display.....	01
Figure 2: QR in Daily Use.....	02
Figure 3: Growing Popularity of QR Codes.....	03
Figure 4: QR Code Related Attacks.....	06
Figure 5: SDGs Adressed.....	11
Figure 6: Block Diagram of QR Shield application.....	22
Figure 7: Steps of Exploitation.....	23
Figure 8: QR Shield Flowchart .....	24
Figure 9: Main Application Display.....	26
Figure 10: Scan from Gallery Display.....	27
Figure 11: Application History Interface .....	28
Figure 12: QR Scan Report .....	29
Figure 13: Use Case Diagram of QR Sheild Application.....	33
Figure 14: Application Permissions.....	35
Figure 15: URL Extraction.....	36
Figure 16: Detailed Analysis of URL .....	37

## List of Tables

Table 2.1: Applications of QR Codes in Various Industries.....	15
Table 2.2: Benefits and Challenges of QR Code Adoption.....	15
Table 2.3: Overview of existing QR Code Scanning Application.....	16
Table 2.4: Common Drawbacks of existing QR Code Scanning Application .....	18
Table 2.5: Innovative Approaches in QR Code Security Research.....	21

## Chapter 1: Introduction

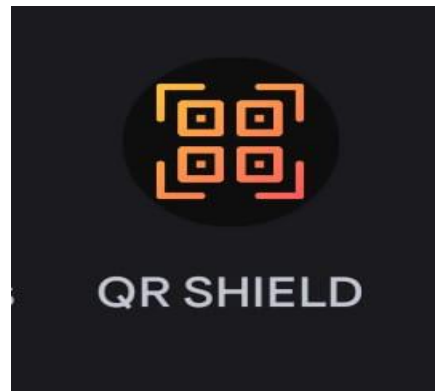


Figure 1: QR Shield Main Display

The rising utility of Quick Response (QR) codes in our regular routines has prompted their far-reaching reception across different businesses, offering efficiency and convenience in accessing cyber content. In any case, the ascent of QR codes additionally delivers security challenges, as malicious agents exploit them for malware distribution, phishing scams, and other cyber threat examples highlights the pressing need for robust security measures to defend clients against potential threats like QR code phishing, malware dissemination, and link spoofing. Moreover, the mix of QR codes into installment frameworks and confirmation processes enhances the earnestness for tough security conventions. As financial exchanges and delicate data trade become progressively dependent on QR innovation, guaranteeing the respectability of these codes becomes vital to obstructing cyber assaults and shielding client data. And furthermore, the multiplication of QR code use in areas like medical care, transportation, and retail highlights the requirement for normalized security practices to moderate possible weaknesses across assorted applications. With QR codes filling in as doors to individual wellbeing records, tickets, and item data, any split difference in their security could have sweeping results, compromising protection, financial stability, and, surprisingly, public wellbeing.

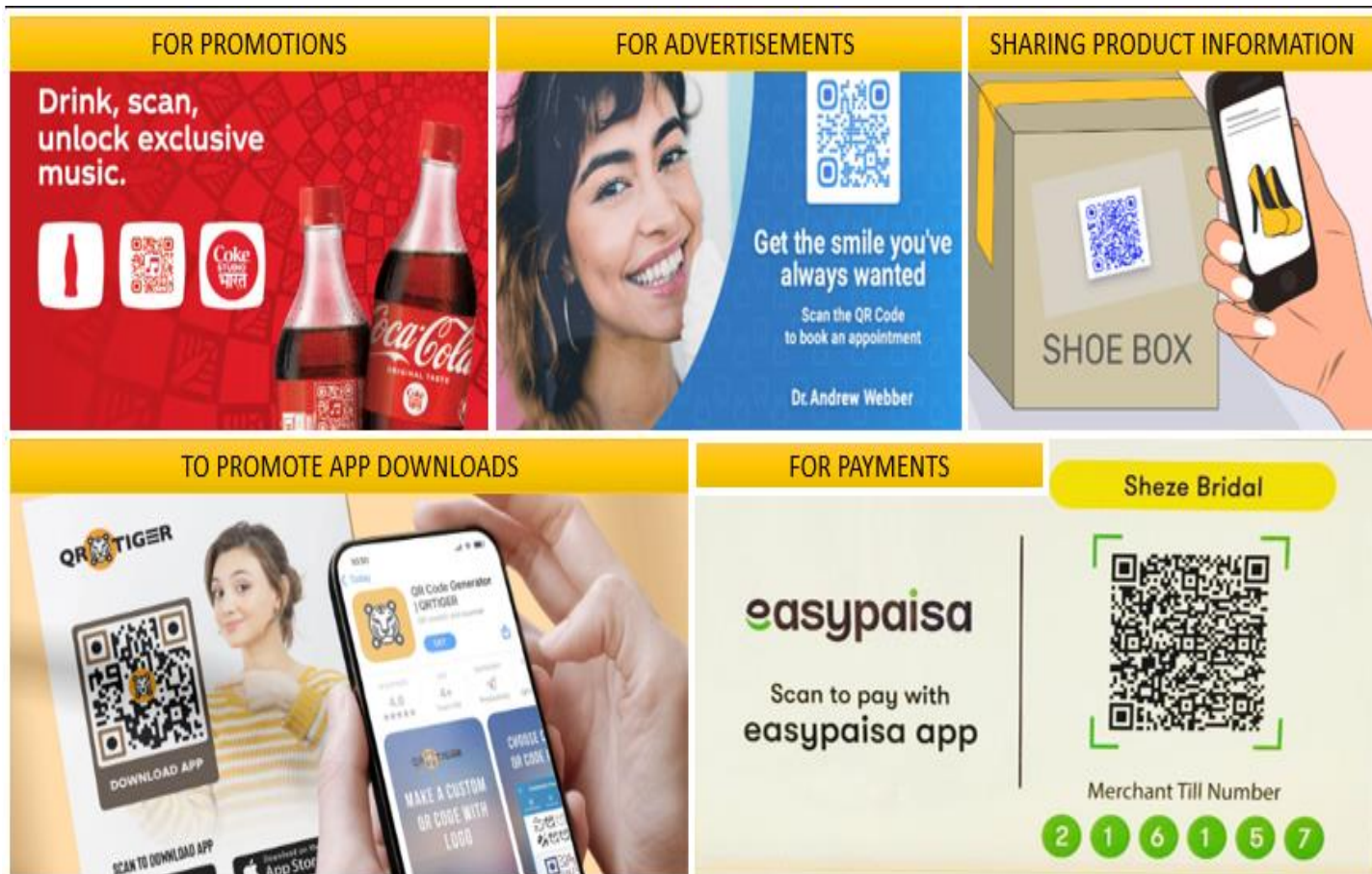


Figure 2: QR in Daily Use

Because of these difficulties, this postulation presents an innovative Android application "QR SHIELD," that pointed toward improving mobile security through real-time threat identification for QR codes. QR SHIELD empowers the clients to survey the security status of filtered QR codes immediately by utilizing progressed examining methods and incorporates consistently with VirusTotal's broad data base of known threats. By making clients aware of potential security gambles, QR SHIELD mitigates the inborn threats related with QR code communications, subsequently engaging the people to explore the advanced scene with certainty. By researching the origination, application, and appraisal of QR SHIELD, this proposal plans to add to the continuous talk encompassing mobile security and QR code examination. This task plans

to further develop client awareness despite arising cyber risks by tending to the security ramifications of QR codes and offering a serviceable arrangement looking like QR SHIELD.

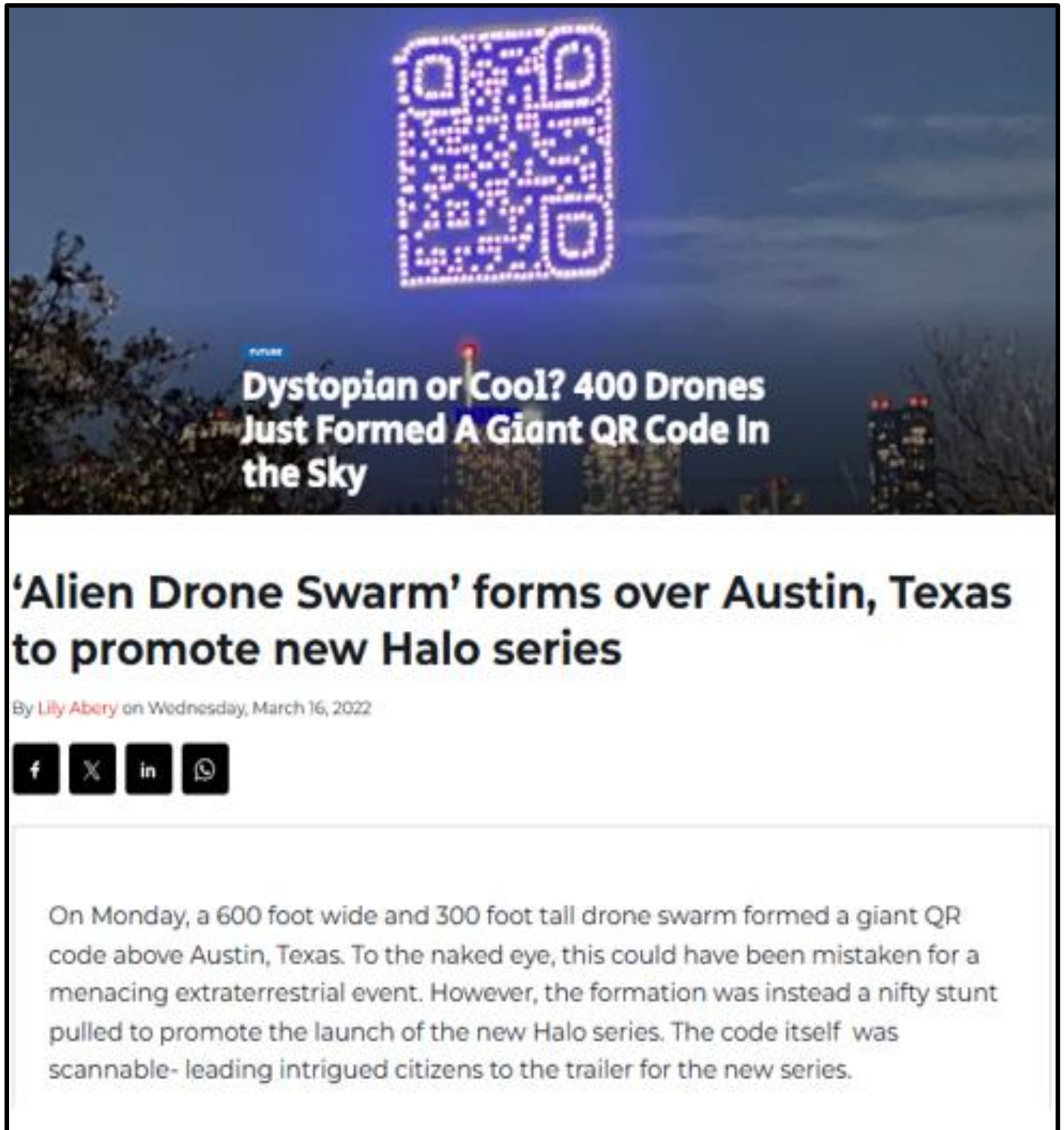


Figure 3: Growing Popularity of QR Codes

## 1.1 Overview

QR codes have become pervasive, going about as easy-to-use tools for data access and exchange consummation. In any case, with their rising popularity additionally comes the chance of malicious QR codes that can think twice about security of clients' gadgets and individual data. Using complex filtering philosophies and joining with Virus Total's huge data set, QR SHIELD tries to outfit clients with brief data in regard to security status of checked QR codes, subsequently working with all around

informed navigation and moderating any threats. The creation, application, and evaluation of QR SHIELD are analyzed in this proposal, giving understanding into its likely applications to work on portable security in the digital period.

## 1.2 Problem Statement

QR codes have turned into a fundamental piece of regular daily existence, working with assortment of errands, such as visiting sites, trading contact data, and making installments. Nonetheless, the expansion in the use of QR codes has likewise gotten an ascent in malicious exercises taking advantage of their usefulness, and elements. Most of the time, clients know nothing about the potential security chances related with examining QR codes, which leave



them helpless against different threats including malware proliferation, phishing plans, and false site redirections. Existing QR code checking applications need robust security measures, subsequently neglecting to shield the clients from these advancing threats. Thus, there is a dire requirement for a solid arrangement that can offer ongoing threat location for QR codes, ensuring the security of clients' gadgets and individual data in an undeniably digitized world.

Conventional QR code scanners basically translate the data without surveying its security, allowing clients to remain uncovered to possible dangers. To moderate these risks, high level QR code scanners outfitted with ongoing danger recognition and confirmation instruments are fundamental. These scanners can dissect the URL or content implanted in the QR code against a data set of known malignant destinations and ways of behaving, hailing or hindering dubious movement before any mischief should be possible. Also, incorporating computerized reasoning and AI can improve the capacity to identify new and developing dangers by perceiving designs demonstrative of noxious expectation. Instructing clients on the dangers related with QR codes and it is similarly critical to advance safe filtering rehearses. Safety efforts, for example, sandboxing, where filtered content is executed in a controlled climate, can additionally safeguard clients. Coordinated effort between network safety specialists and QR code innovation engineers is vital to remain in front of cybercriminals. As the computerized scene keeps on advancing, focusing on the security of QR code collaborations will assist with protecting individual data and keep up with trust in advanced exchanges.

On the next page a snapshot of latest QR code attacks and their changes frequency of attacks is displayed based on some news and articles by security analysts explaining the gravity of the issue at hand. These growing attacks motivated us to choose this project to create a more safer mobile devices environment.



IMAGE: MITYA IVANOV VIA UNSPLASH

Jonathan Greig  
August 17th, 2023


Cybercrime Industry

### Phishing campaign used QR codes to target large energy company

Cybersecurity researchers uncovered a large phishing campaign using malicious QR codes with the hopes of acquiring Microsoft credentials at several targets, including a major U.S. energy company.

COFENSE  
EMAIL SECURITY

Stop Threats Solutions Clients Resources About



Marketing String

Victim's email

Base64 Encoded Phishing Link

Figure 2: Bing Redirect URL

Over 2,400% Increase in Malicious QR Code Phishing Volume

NBC BAY AREA LOCAL WEATHER INVESTIGATIONS VIDEO SPORTS NEWSLETTERS

TRENDING Biden in Bay Area Power bills Illegal firearms arrest Rent prices Bullying case Donald Trump Deals for You

NBC BAY AREA RESPONDS

### Fake QR Codes Can Expose Your Phone to Hackers. to Protect It

QR codes are more popular than ever with businesses, as they offer a convenient and touchless way to share information. But bad actors can replace QR codes in public with their own, granting them access to your phone.

By Chris Chmura and James Jackson • Published October 4, 2020 • Updated on October 6, 2020 at 8:18 am

Technology MAGAZINE

News & Articles Magazines Reports & Whitepapers Sectors Events

Article • Cloud & Cybersecurity

### Report Highlights Rising Threat of C-Suite QR Code Attacks

By Marcus Law  
February 07, 2024 • 4 mins

### Warning over QR code scams after 'quishing' victim loses £13,000 to scammers

ANGLIA | CRIME AND COURTS | CAMBRIDGESHIRE POLICE | CYBER CRIME

Wednesday 24 January 2024 at 6:00am



A new poster campaign by Cambridgeshire Police aims to alert people to the risks of QR code scams.  
Credit: Cambridgeshire Police

### Bengaluru: QR code fraud makes up 40% of cases since 2017

Rajiv Kalkod / TNN / Updated: Aug 21, 2023, 12:37 IST

SHARE AA FOLLOW US

A 30-year-old Indian Institute of Science professor, who sought to sell his washing machine on an online marketplace on August 11, lost over Rs 60,000 to cybercriminals.

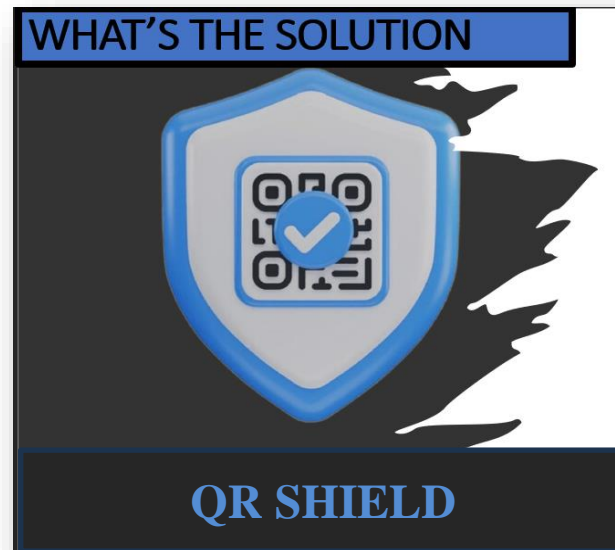


File photo for representation

Figure 4: QR Code Related Attacks



### 1.3 Proposed Solution



To address the security concerns presented by malicious QR codes, this postulation proposes the improvement of "QR SHIELD," an inventive Android application intended to upgrade portable security by provoking clients about the noxiousness and different qualities of the sites they plan to visit. QR SHIELD utilizes the proficient examining strategies to dissect the substance of filtered QR codes continuously, and afterward distinguishing potential threats including noxious URLs, phishing endeavors, and malware payloads QR SHIELD gives clients quick data about the security status of checked QR codes, empowering them to make educated, effective choices while cooperating with digital content by incorporating itself with Virus Total's broad data set of known threats. Also, QR SHIELD consolidates a high-level threat ready framework that will tell the clients of distinguished threats, in this manner enabling buyers to jump into the advanced world with strength. QR SHIELD means to moderate the threats related with QR code cooperations to improve the client wellbeing in an undeniably interconnected world through its exhaustive way to deal with QR code security

## 1.4 Working Principle

QR SHIELD works on a straightforward however critical standard: constant threat location for QR codes. At the point when a client examines a QR code utilizing the QR SHIELD application, it starts a Quick checking process, which will break down the code content progressively which includes parsing the encoded data inside the QR code and alluding to VirusTotal to evaluate its wellbeing status. QR SHIELD analyzes the substance that has been examined for indications of perilous action, like questionable URLs, known malware marks, and phishing endeavors using the Virus Total's database.

The fundamental component of QR Shield's operation is its integration with Virus Total's extensive threat intelligence platform. When a QR code is scanned, QR Shield sends the content of the code to Virus Total's database, where it is carefully examined against a large database of known threats. VirusTotal uses a variety of detection approaches, such as signature-based scanning, heuristic analysis, and behavioral tracking to identify and categorize potential threats accurately.

Once VirusTotal completes its analysis, the QR Shield receives the results and prompt the users about the security status of the scanned QR code. If the code is flagged as malicious or suspicious, QR Shield alerts the user along with the details about the detected threat and in the case of legitimate code the user would still be notified and will proceed with his own will thus adding on to the security status of the internet usage, personal data and mobiles security.

By using the efficient scanning techniques and its integration with reputable threat intelligence source, VirusTotal, QR Shield ensures that users can scan QR codes without any having to worry about security, confirming that security is prioritized at every step of the process. This proactive approach to QR code security empowers users to make informed and safe decisions when diving

into the cyber world, thus mitigating the risks associated with QR code usage and adding to the mobile security efficiently.

## **1.5 Objectives**

### **1.5.1 General Objectives:**

- a. Enhancing mobile security by providing real-time threat detection for QR codes.
- b. Empowering users with safe and efficient decision-making capabilities when scanning QR codes.
- c. Mitigating the QR code-related threats like malware distribution and phishing.
- d. Increasing the user confidence and trust through a transparent, safe and reliable solution.
- e. Contributing to the advancements in the research of mobile security.

### **1.5.2 Academic Objectives:**

- a. Examining the security risks that QR codes present and how they affect mobile users.
- b. Evaluating the application and performance of QR Shield in real-world scenarios.
- c. Contribute research findings and thoughts on mobile security and QR code analysis to scholarly conversation.

## **1.6 Scope**

The extent of QR SHIELD incorporates all phases of the improvement interaction for a high-level Android application intended to further develop QR code security. This involves making and setting in motion advances like ongoing checking, and mix with outsider threat knowledge administrations like VirusTotal. Broad testing conventions are executed to check the application's adequacy in recognizing and turning away threats related with QR codes, ensuring max operation in down to earth circumstances. All through the improvement cycle, the client's experience is given main concern, with specific spotlight on availability and easy to understand interface plan.

Moreover, broad documentation is made to make it simpler to grasp and share QR SHIELD's outcomes and capacities, ensuring its appropriateness and viability in further developing client security while managing QR codes on Android gadgets.

## **1.7 Deliverables**

### **1.7.1 QR Shield Application:**



The essential deliverable of the task is the Android application, QR SHIELD, furnished with elements, for example, ongoing examining, threat identification, ready warnings to provoke clients, and safe perusing client's will.

### **1.7.2 Technical Documentation:**

Complete documentation will be given which will detail the plan, usefulness, and execution of the QR SHIELD application. This documentation will likewise act as a kind of perspective various partners engaged with the venture.

### **1.7.3 User Manual:**

The QR Shield application will come with user-friendly documentation that will walk users through the installation, configuration, and successful use of the program. These handbooks will improve the application's uptake and user comprehension.

### **1.7.4 Source Code:**

The QR Shield application's source code will be made accessible, thus promoting openness, cooperation, and potential future improvements by the developer community.

## 1.8 Relevant Sustainable Development Goals



Figure 5: SDGs Addressed

The project addresses the local demand for secure infrastructure by developing a system to recognize and flag harmful QR codes, thus promoting security and reliability in QR code usage, thereby contributing to *SDG 9: Industry, Innovation, and Infrastructure*. The project also contributes to *SDG 16: Peace, Justice, and Strong Institutions* by addressing the local socio-economic issue of ensuring security and shielding people and companies from fraud and cyber threats through the implementation of a system to recognize and label malicious QR codes.

## 1.9 Structure of Thesis

- **Chapter 2** contains the literature review and the background and analysis study this thesis is based upon.
- **Chapter 3** contains the design and development of the project.
- **Chapter 4** introduces detailed evaluation and analysis of the code.
- **Chapter 5** contains the conclusion of the project.
- **Chapter 6** highlights the future work needed to be done for the commercialization of this project.

## Chapter 2: Literature Review

In the improvement of QR SHIELD, a broad literature review expects a fundamental part in teaching the cognizance in regards to existing examination, game plans, and troubles connecting with QR code security. By on a very basic level taking a gander at prior works, we can perceive openings in data, anticipated locales for improvement, and imaginative ways of managing address security concerns related with QR code use. Hence, this literature audit fills in as the essential design whereupon QR SHIELD is built, ensuring that it is taught by the latest types of progress and encounters in the field. And also by analyzing existing studies methodically, we can benchmark QR SHIELD against the methodologies and best practices in the field. This comprehensive review guides the development of robust, cutting-edge security features for QR SHIELD in addition to identifying existing vulnerabilities .

The development of this literature review segment is expected to give an intentional examination of key viewpoints associated with QR code innovation and security. We start by diving into the modern foundation of QR code innovation, following its turn of events, applications across various undertakings, and the security ideas in that (Chang et al., 2020; Chen et al., 2021; Thoughtful et al., 2020). Understanding the modern scene is basic for contextualizing the progression of QR SHIELD and seeing the crushing requirement for solid wellbeing endeavors in QR code joint efforts.

Then, we separate existing game plans and their drawbacks in QR code sifting applications, drawing pieces of information from Ongoing assessments and outlines (Alqahtani and Liu, 2022; Gupta et al., 2021; Zhang et al., 2019). This assessment empowers us to recognize normal limits, for example, lacking threat recognition abilities and convenience issues, which QR SHIELD means to address through inventive plan and usefulness.

At last, we review Recent research papers zeroing in on QR code security, featuring arising patterns, techniques, and possible arrangements (Fu et al., 2022; Ahmed et al., 2021; Chen et al., 2020). By blending bits of knowledge from these examinations, we plan to use state of the art ways to deal with improve the adequacy and strength of QR SHIELD in defending clients against advancing threats.

In outline, this literature review section fills in as a thorough investigation of the current information scene encompassing QR code security, laying the background for the ensuing plan and development periods of QR SHIELD.

## **2.1. Industrial Background of QR Code Technology**

The industrial background of QR code technology traverses many years, described by its development from a specialty creation to a pervasive instrument in different areas. QR codes, or Fast Reaction codes, were first imagined by Denso Wave, an auxiliary of Toyota, in 1994, to follow car parts during assembling (Denso Wave Inc., 2023). At first, QR codes filled utilitarian needs in industrial settings, offering an additional productive method for information capacity and recovery contrasted with customary standardized tags.

Notwithstanding, the capability of QR codes before long stretched out past the industrial facility floor as their special capacities earned respect. By encoding data in both vertical and even examples, QR codes could store fundamentally a greater number of information than customary standardized tags, prompting their reception in different ventures. In the mid-2000s, QR codes started showing up in showcasing efforts and ads, permitting purchasers to get to extra item data or limited time content by checking codes with their cell phones.

The far and wide multiplication of cell phones in the last part of the 2000s further sped up the reception of QR codes, changing them into flexible apparatuses for connecting the physical and

advanced universes. Retailers utilized QR codes to work with versatile installments, empower item confirmation, and improve client commitment through intuitive encounters (Chen et al., 2021). In the transportation area, QR codes upset tagging frameworks, permitting travelers to get to tickets, travel timetables, and course data with a straightforward sweep.

In medical services, QR codes arose as significant apparatuses for smoothing out persistent consideration processes, empowering clinical experts to get to electronic wellbeing records, prescription data, and therapy conventions in a hurry (Chang et al., 2020). In like manner, QR codes found applications in operations and store network the board, working with stock following, item confirmation, and shipment check (Chen et al., 2021).

While the reception of QR codes acquired evident advantages terms of proficiency, accommodation, and upgraded buyer encounters, it likewise raised worries about security and protection. QR codes, by their actual nature, can contain erratic information, including URLs, contact data, and touchy individual information. In that capacity, vindictive entertainers have taken advantage of QR codes for different detestable purposes, including phishing tricks, malware conveyance, and fake exercises (Goodness et al., 2020).

The powerful idea of QR code content presents difficulties for clients and associations the same, as it turns out to be progressively challenging to recognize authentic and malignant codes. Besides, the consistent mix of QR codes into ordinary exercises, combined with the pervasiveness of QR code checking applications, has made a ripe ground for digital threats to multiply (Goodness et al., 2020). Thus, there is a squeezing need for upgraded safety efforts to relieve the threats associated with QR code utilization and SHIELD clients' gadgets and individual data. Thus, The implementation of advanced security measures like QR SHIELD is beneficial in safeguarding users and organizations against these evolving advanced and legacy threats.



**Table 2.1: Applications of QR Codes in Various Industries**

<b>Industry</b>	<b>Applications</b>
Retail	Mobile payments, product authentication, marketing
Transportation	Ticketing, boarding passes, transit information
Healthcare	Electronic health records, medication information
Logistics	Inventory tracking, shipment verification
Advertising	Promotional campaigns, interactive experiences

**Table 2.2: Benefits and Challenges of QR Code Adoption**

<b>Benefits</b>	<b>Challenges</b>
Efficiency	Security risks
Convenience	Privacy concerns
Enhanced consumer experiences	Malicious exploitation
Seamless integration into existing systems	Difficulty in distinguishing legitimate codes

In outline, the industrial background of QR code technology mirrors its excursion from a specialty industrial device to a universal presence in different areas. While QR codes offer various

advantages as far as effectiveness and comfort, their broad reception has likewise raised huge security concerns. Tending to these difficulties requires a complete methodology that coordinates upgraded safety efforts with the proceeded with development and advancement of QR code technology.

**2.2. Existing Solutions and Their Drawbacks**

Existing QR code scanning applications play a crucial role in enabling users to interact with QR codes and access digital content seamlessly. Nonetheless, a basic examination uncovers a few limits and deficiencies that thwart their viability in guaranteeing client security and protection.

**2.3. Overview of Existing QR Code Scanning Applications:**

**Table 2.3: Overview of Existing QR Code Scanning Applications**

<b>Application</b>	<b>Features</b>	<b>Limitations</b>
QR Scanner	Basic scanning functionality	Limited threat detection capabilities
Barcode Scanner	QR code scanning, barcode recognition	Minimal user guidance, lack of updates
QR Code Reader	Fast scanning speed, simple interface	Reliance on static blacklisting approaches
Scan Life	QR code scanning, product information access	Inconsistent performance, usability issues

#### **2.4. Identification and Discussion of Limitations:**

Many existing QR code checking applications experience the evil impacts of basic limitations that compromise client security and insurance. One typical detriment is the lacking threat distinguishing proof capacities of these applications. While they may really look at QR codes and unravel their things, they habitually disregard to take apart the checked data for potential security bets. Likewise, clients may accidentally open themselves to harmful substance, including phishing destinations, malware payloads, and underhanded plans (Alqahtani and Liu, 2022).

Additionally, existing QR code checking applications could require enthusiastic client course and tutoring on potential security bets. Clients are a significant part of the time left to investigate QR code correspondences isolated, without clear rules or cautions about probable threats. This shortfall of heading works on the likelihood of clients surrendering to phishing stunts or coincidentally downloading malware onto their contraptions (Gupta et al., 2021).

Another basic limitation is the reliance on static systems for threat area. Various QR code analyzing applications utilize clear boycotting or whitelisting procedures to perceive vindictive URLs or known threats. Regardless, these static strategies are naturally confined in their sufficiency, as they rely upon pre-described courses of action of known threats and may fail to perceive emerging or ahead of time dark threats (Zhang et al., 2019).

Moreover, the shortfall of ongoing updates and versatile learning in numerous QR code examining applications leaves them unprepared to handle advancing digital dangers. These applications frequently don't integrate AI calculations that could upgrade their aggressive statement location abilities by gaining from new information over the long haul (Kim et al., 2020). Besides, some QR code scanners need reconciliation with more extensive security structures, which could give more far-reaching insurance by cross-referring to identified dangers with a bigger data set of

known chances (Lee and Park, 2018). The restricted client mindfulness and commitment highlights in these applications further worsen the gamble, as clients are not adequately informed about safe QR code utilization rehearses (Wang et al., 2021). Ultimately, protection concerns emerge when QR code examining applications gather and store client information without satisfactory shields, possibly presenting delicate data to unapproved access or abuse (Chen and Liu, 2021).

**2.5. Examples of Common Drawbacks:**

**Table 2.4: Common Drawbacks of Existing QR Code Scanning Applications**

Drawbacks	Implications
Inadequate threat detection capabilities	Increased risk of malware infection
Lack of user guidance	Higher susceptibility to phishing scams
Reliance on static approaches	Inability to detect emerging threats
Usability issues	Decreased user confidence and trust

**2.6. Review of Relevant Studies and Research Papers:**

A couple of examinations and research papers have highlighted the hardships looked by existing QR code sifting applications and the implications for client security and insurance. Alqahtani and Liu (2022) coordinated an outline on the security shortcomings of QR code applications, uncovering enormous lacks in threat revelation limits and client preparing. Furthermore, Gupta et al. (2021) recognized security shortcomings and proposed countermeasures

to work on the security of QR code separating applications. Zhang et al. (2019) drove a total report on attacks and SHIELDS on QR codes, underlining the restrictions of static systems and the necessity for dynamic assessment procedures.

### **2.7. Discussion on Implications for User Security and Privacy:**

The limitations and insufficiencies of existing QR code checking applications have basic consequences for client security and assurance. Lacking threat ID limits increase the bet of clients unintentionally introducing themselves to malevolent substance, inciting potential data breaks, money related incidents, and information misrepresentation. Nonappearance of client course and guidance further demolishes these threats by leaving clients vulnerable against social planning attacks and phony activities. Reliance on static procedures limits the reasonability of threat acknowledgment and fails to address emerging threats, introducing persistent challenges for client security (Zhang et al., 2019).

With everything taken into account, current QR code separating applications face different limitations and shortcomings that compromise client security and assurance. Keeping an eye on these troubles requires a comprehensive philosophy that directions advanced threat distinguishing proof methodologies, client preparing, and dynamic assessment procedures. By easing these disservices, future QR code looking at applications can more probably shield clients from creating computerized threats and assurance a safer electronic understanding.

### **2.8. Research Papers on QR Code Security**

Recent advancements in QR code security have nudged a creating gathering of assessment highlighted working on the confirmation of clients against emerging threats and shortcomings. This section gives a review of select assessment papers and studies focusing on QR code security, highlighting creative systems and techniques proposed in the literature.

## 2.9. Overview of Innovative Approaches and Methodologies:

Machine Learning-Based Threat Detection: Ahmed et al. (2021) proposed a AI based approach for phishing ID in QR codes. By using artificial intelligence computations, the audit achieved predominant accuracy in recognizing poisonous QR codes, appropriately further developing client security and mitigating the bet of surrendering to phishing stunts.

Dynamic Assessment Procedures: Chen et al. (2020) introduced the possibility of dynamic QR codes as a unique method for managing overhauling security. Dynamic QR codes engage nonstop threat assessment by combining dynamic parts like time-fragile data or confirmation instruments. This approach works on the strength of QR codes against poisonous control and ensures that clients get cutting edge information while restricting the bet of receptiveness to threats.

Mix with Threat Information Stages: Fu et al. (2022) researched the mix of QR code inspecting applications with threat understanding stages to further develop security. By using ceaseless threat data and helpful assessment from different sources, the review showed additionally created precision in perceiving and mitigating QR code-associated threats. Mix with threat information stages outfits QR code sifting applications with permission to a wealth of threat understanding, engaging proactive threat area and response.

In overview, research papers focusing in on QR code security have introduced imaginative approaches and methods highlighted further creating client security and easing the threats associated with QR code use. By using artificial intelligence-based threat revelation estimations, dynamic examination procedures, and getting together with threat information stages, QR SHIELD can update its disturbing assertion area limits and give clients a more secure QR code looking at experience. **So**, the literature review has given important bits of knowledge into the present status of QR code security, featuring both the open doors and difficulties associated with QR code utilization.

**Table 2.5: Innovative Approaches in QR Code Security Research**

<b>Research Papers</b>	<b>Approach/Methodology</b>	<b>Findings/Implications</b>
<b>Ahmed et al. (2021)</b>	Machine learning-based phishing detection for QR codes	Improved detection accuracy for malicious QR codes
<b>Chen et al. (2020)</b>	Dynamic QR code: A novel approach to enhance security	Real-time threat assessment and adaptive QR codes
<b>Fu et al. (2022)</b>	Enhancing QR code security with threat intelligence	Integration with threat intelligence platforms

Key discoveries incorporate the broad reception of QR codes across different businesses, the impediments of existing QR code checking applications, and creative methodologies proposed in ongoing research papers to improve QR code security. These bits of knowledge advise the development regarding QR SHIELD by underlining the significance of powerful threat recognition abilities, client direction, and joining with cutting edge security advancements.

### 3.1 Design Process

The layout system for the QR Shield application contains many steps of a two-fold definition which targets specific components of the consumer interface and user experience.

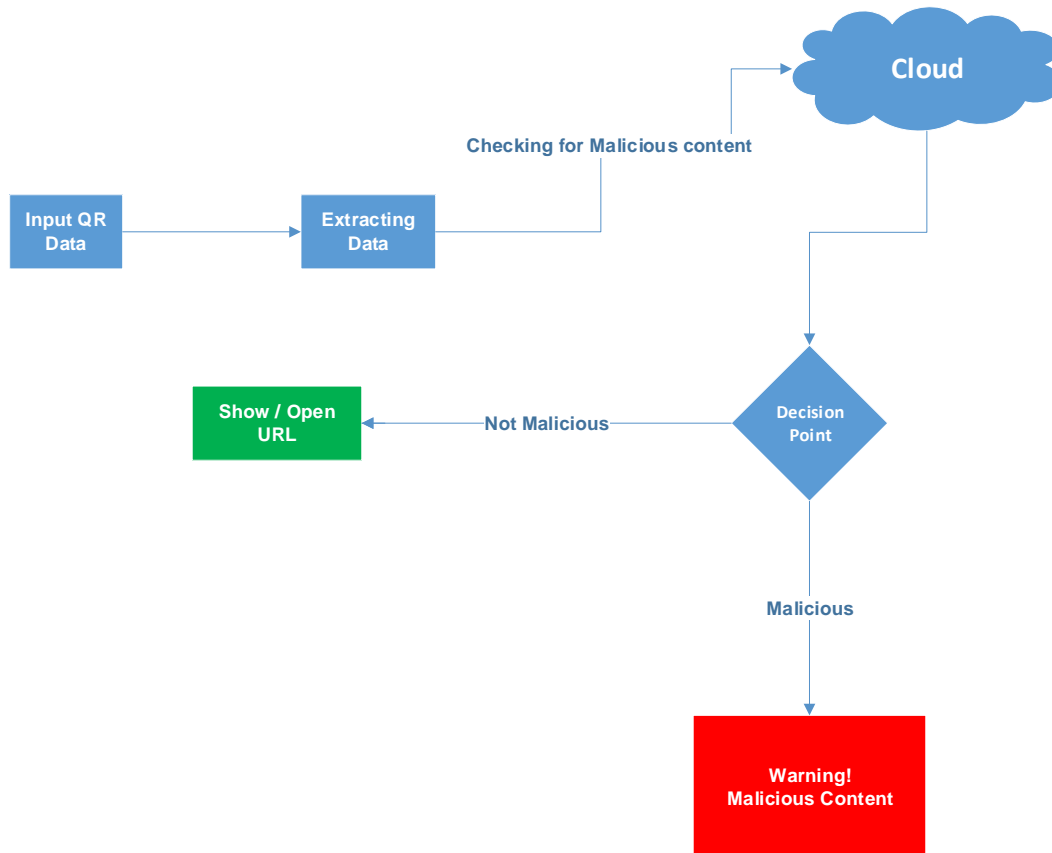


Figure 6: Block Diagram of QR Shield application

#### 3.1.1 User Research

In making the app, we have established the essential studies such as the behavior surveys, interviews, usability tests, and the checking of the possibilities and capability of the phones. The personas that we anticipate being our users and the requirements that relate to them regarding QR code scanning and security have already been identified.



### 3.1.2 Information Architecture

We have given details and shown a view of how the program will look like having done our due diligence. Two primary features of the app are the capacity to scan QR code, historical scans, integration of VirusTotal using its API key, settings and alerts that can be controlled. Based on their importance and interdependence, we determined the relationships between these components and establish a hierarchy. The structure of the users' interactions and movements in sequence was taken into account when the documents were classified according to their categorization and expected use order.

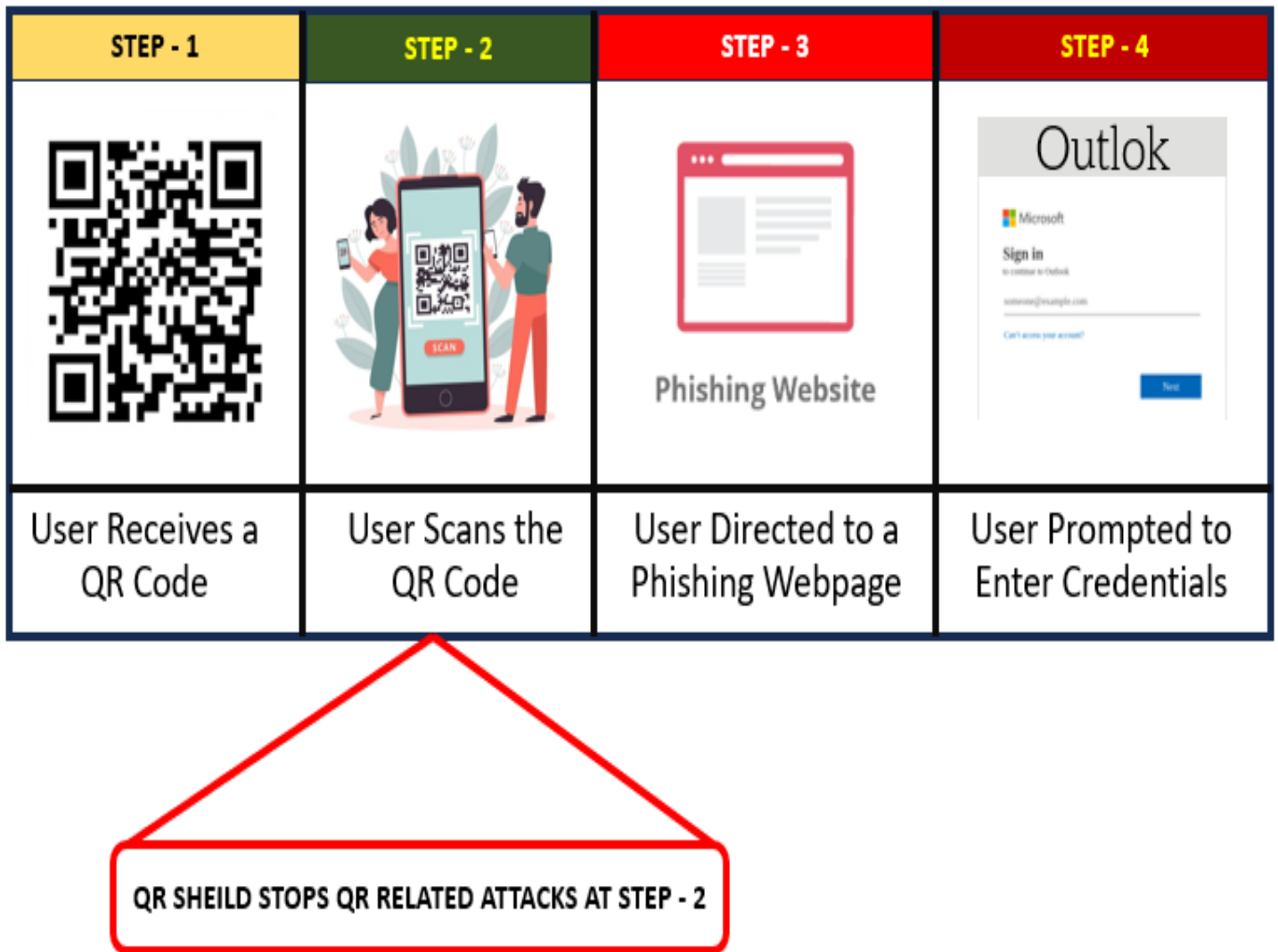


Figure 7: Steps of Exploitation

### 3.1.3 Flowchart Diagram

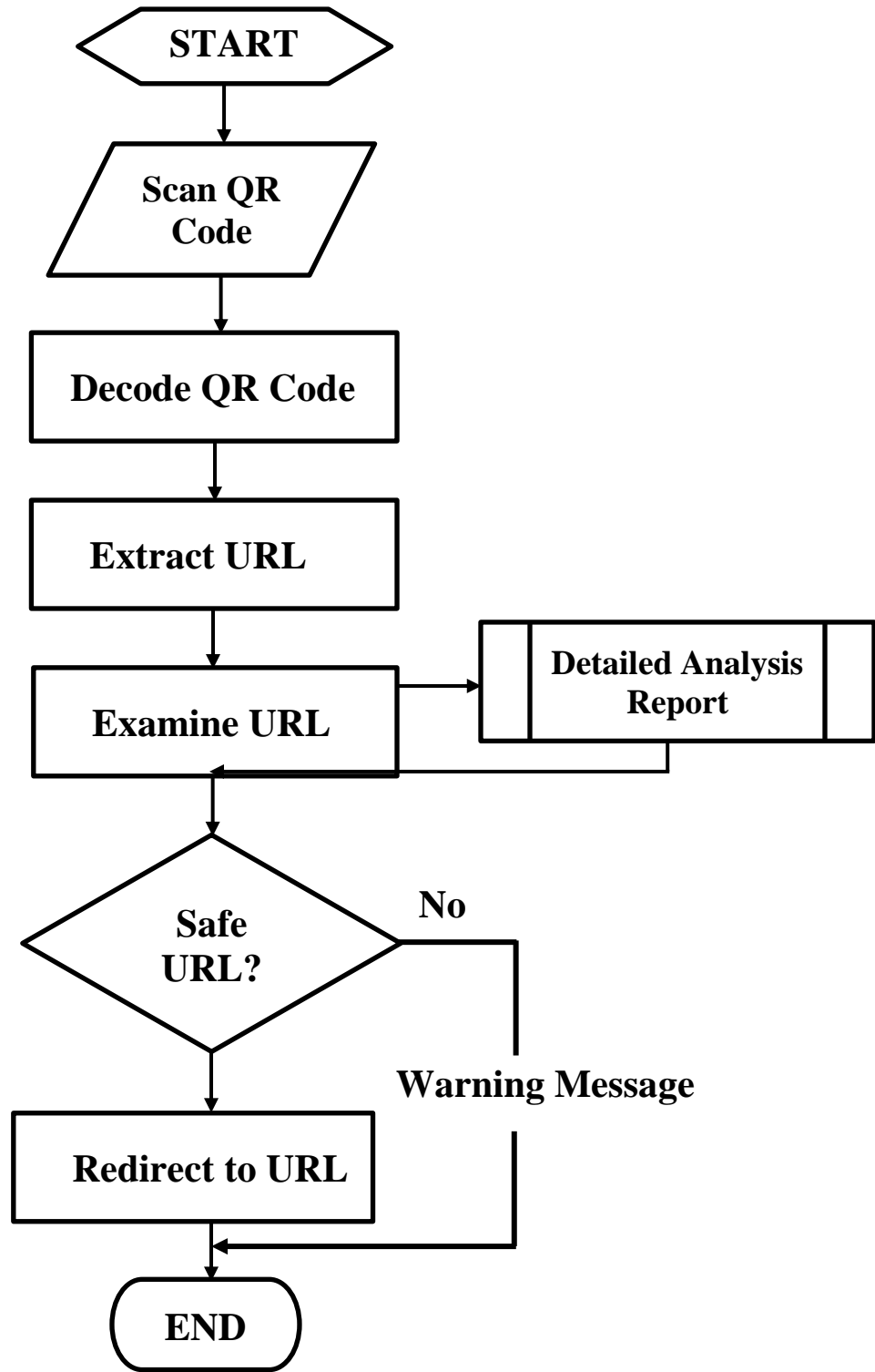


Figure 8: QR Shield Flowchart

### **3.1.4 Wireframing**

The interaction experience was first mocked-up in wireframes that employed low fidelity for the sake of seeing the format and shape of the utility. The attention was specially paid to the placement of those mechanisms together with QR code scanner, experiment history view, parameters, and alerts.

### **3.1.5 Prototyping**

Once the wireframe was ready, then an interactive prototype was created using the Adobe XD for the user interface. We utilized our present prototype and tested with customers to gather their feedback and make important changes.

### **3.1.6 Visual Design**

Visual design elements such as colors, images and typography have been added to enhance the design of the prototype after ensuring that the prototype was both usable and functional. We made sure that the design aspects were consistent and that they were adequately aligned with the application's branding standpoint.

### **3.1.7 High-Fidelity Prototyping**

High-fidelity prototypes were created with improved visual designs and realistic content. These prototypes closely resemble the final product and are used for final testing and validation before development begins.

### **3.1.8 Usability Testing**

We did usability studies using actual customers to ensure app design and its functionality and to pinpoint any usability issues. Design was changed based on the customer's comments and system checking results to make the user's experience better.

## **3.2 User Interface Features**

### **3.2.1 QR Code Scanning Display**

We modeled a convenient and user-friendly interface for QR code scanning. It involves clean commands, comments, and messages that users use in articulating the purpose of the process. The main screen of an application typically shows the scanning

interface and all interactions are made through it by users with the QR codes. It has some scanning options such as the QR code scanning or from gallery. Besides, this interface also provides an option to access history.

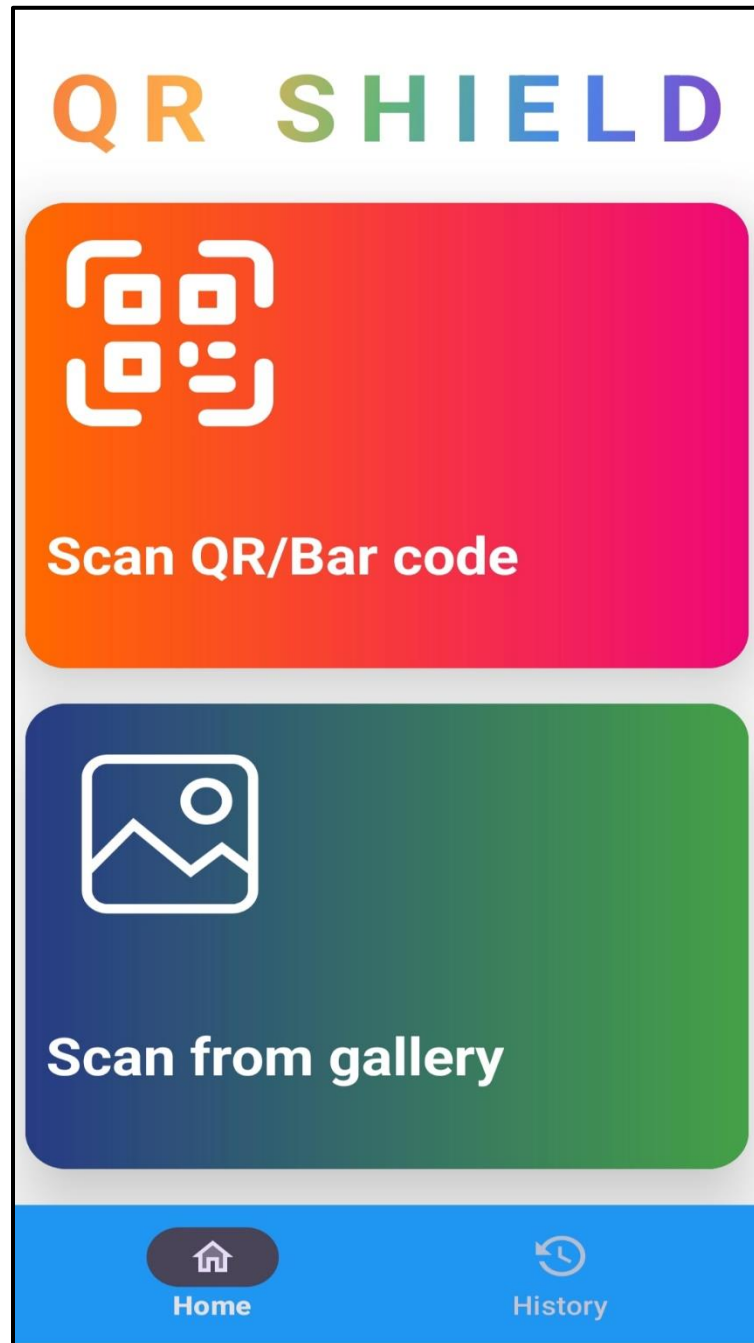


Figure 9: Main Application Display

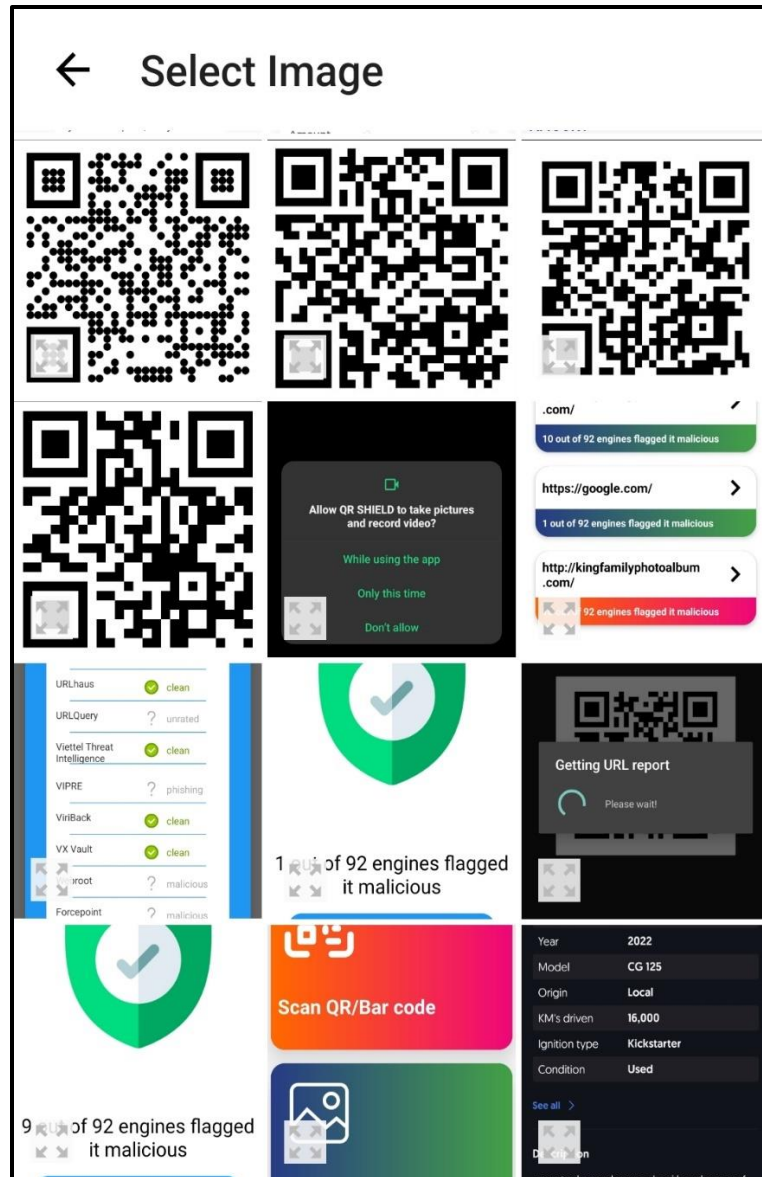


Figure 10: Scan from Gallery Display

### 3.2.2 Scan History Display

This interface is typically accessible from the main screen of the application. It displays a chronological list of all scanned QR codes using the information consisting of the URL and number of security engines which flagged it malicious. Applying filters and sorting options to effortlessly navigate and control experiment records.

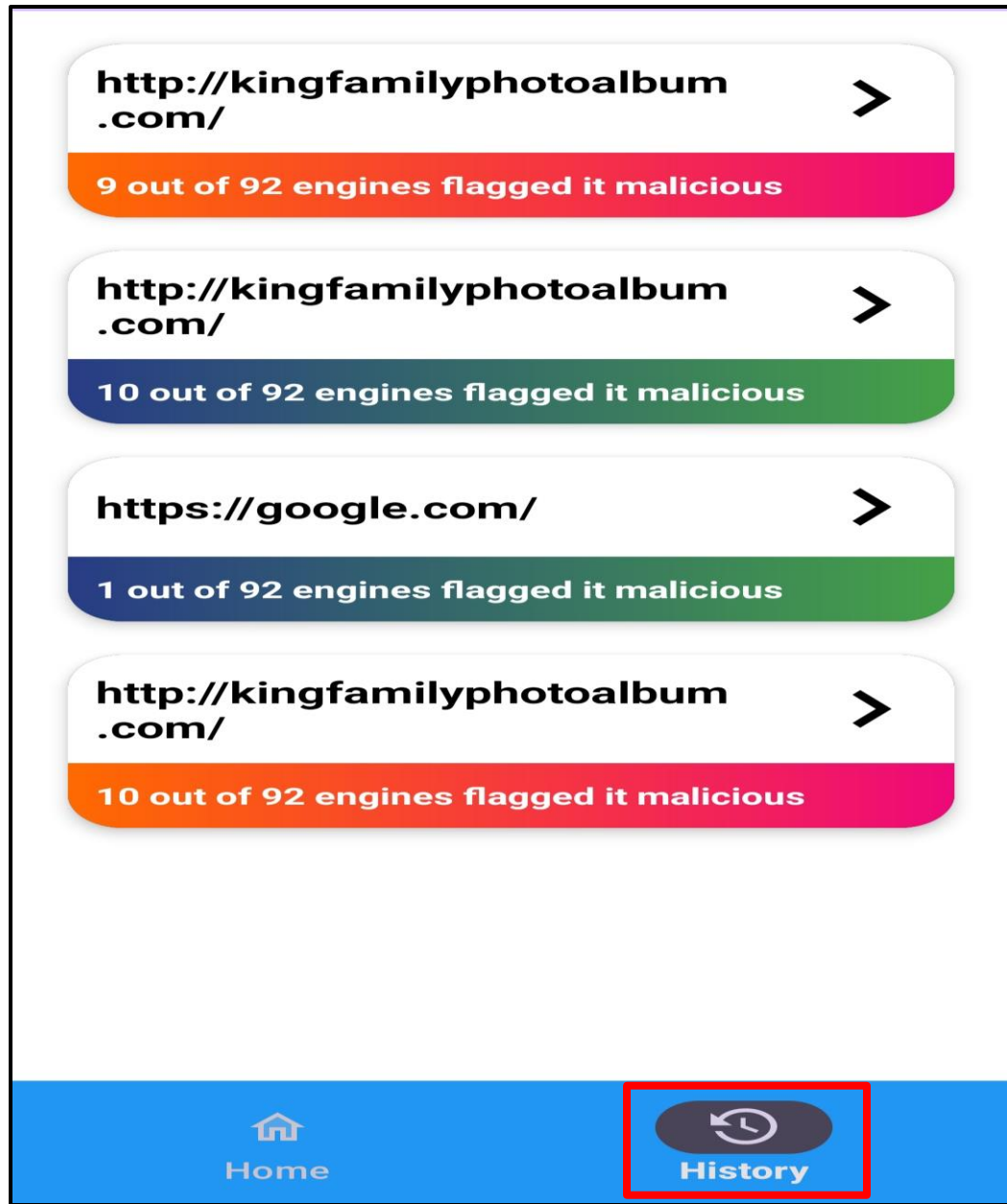


Figure 11: Application History Interface

### 3.2.3 Alert Notifications Display Screen

QR code-scanning notification system was created for detecting customers numerous fraud charges and other harmful activities. It closely mirrors the real-world scenario which acts as an effective aid when performing test practices. It also has test report options, along with actions like opening the URL or aborting it.

<http://kingfamilyphotoalbum.com/>



9 out of 92 engines flagged it malicious

Open full report

Click to open URL

Click to abort

Figure 12: QR Scan Report

### 3.3 DEVELOPMENT PROCESS

#### 3.3.1 Requirement Analysis

First, the study was conducted through the existing scanning apps of QR code to understand the market and potential features. Purposeful and non-functional necessities were evaluated and trimmed. We came up with personal stories, use cases and reputation criteria which will be used as the base of the application's development process.

### 3.3.2 Development Phase

#### 3.3.2.1 Frontend Development

For the frontend of the QR Shield application, our focus was on the user interface (UI) and user experience (UX) design, as well as the implementation of QR code scanning functionality. We have used Java programming language for native android development. UI components were Android SKD. For QR scanning, zxing library was incorporated into the codebase. Design elements were applied with Android XML layout controls and completely separated resources.

#### 3.3.2.2 Backend Development

Node.js was used for improving the features in the server side. The application server and database management system were installed in the given environments (MongoDB) as well. The APIs were incorporated to develop the user module, QR code scanning, and VirusTotal integration. The commercial sense of running QR code scan, get results, and send notifications is also put into effect.

#### 3.3.2.3 Integration with Virus Total API

API keys existed on VirusTotal, and everyone can easily get them after he/she registers with it instantly. A QR code requests programmers used to handle this task. API keys existed on Virus Total and everyone can easily get them after he/she registers with it instantly. A QR code requests programmers used to handle this task. They could deliver QR code records to VirusTotal for scanning and retrieve the scan results. This is using http requests calling Virus Total APIs to send the scanned QR code data and get responses if the info in the QR codes is malicious. The program then processes the scan data and weighs results to single out malicious or virus containing QR codes that may potentially endanger the system. The integration with VirusTotal's API enabled seamless automation, allowing the program to continuously monitor and scan QR codes without manual intervention. This approach ensured a rapid response to potential threats, as the program could immediately flag and isolate malicious QR codes based on the real-time scan results from VirusTotal.





### **3.3.3 Testing Phase**

Carrying out the process of testing phase is paramount to establish that QR Shield app is functioning properly, is user-friendly and gives accurate results.

#### **3.3.3.1 Unit Testing**

Tests Unit assessment was developed to verify that functionality of individual modules, components, and capabilities are as required. Jest was used to make test framework for JavaScript.

#### **3.3.3.2 Functional Testing**

##### **3.3.3.2.1 QR Code Scanning**

Finding out how well the app could detect different types of QR codes that were scanned under different lighting conditions was the aim of the research. We made the application to work with QR codes of different sizes and with varying levels of difficulty and complex levels.

##### **3.3.3.2.2 Malicious Content Detection:**

We made sure that the API implemented by VirusTotal API works correctly by identifying a malicious content scan on QR codes. Performed tests both with the help of known harmful and harmless QR codes for evaluating detection efficiency. Additionally, we conducted stress testing to ensure the API's reliability and performance under high-volume scan requests.

##### **3.3.3.2.3 User Interface (UI) Testing**

The interface performance test was designed to be fail-safe and to confirm all the UI parts are well-functioning and responsive. Tested optimum transition between screens and buttons functionality.

##### **3.3.3.2.4 Error Handling:**

The performance of the app in unexpected conditions like network errors, API request rate limiting, or invalid QR codes was verified. When there are mistakes made by the user, the system will display helpful messages to inform the user.

### **3.3.3.3 Compatibility Testing**

#### **3.3.3.3.1 Device Compatibility**

The application was tested on a variety of Android devices to ensure compatibility with different screen sizes, resolutions, and hardware configurations.

#### **3.3.3.3.2 Operating System Compatibility**

The mobile version has been tested on various versions of Android ranging from the older devices to the newer ones to ascertain compatibility with older and newer OS versions. QR Shield is compatible with Android 10 or higher versions. Checks were made, to ensure that the technical part both conforms to the given design principles and as well as the behavior of the platform.

### **3.3.3.4 Performance Testing**

#### **3.3.3.4.1 Scanning Speed**

A check was made that how fast the app detects the QR codes and populates the results. Performance was optimized to avoid any problems with the speed and efficiency of scanning.

#### **3.3.3.4.2 Resource Usage**

We checked CPU, memory, and voltage during scanning process to detect and resolve resources load or battery life shortening issues.

#### **3.3.3.4.3 Integration Testing**

Integration test involved frontend and backend subroutines. We have made how the app is supposed to work end to end, like QR code scanning, VirusTotal integration, and notification alert.

#### **3.3.3.4.4 User Acceptance Testing (UAT)**

We have conducted UAT with the actual user for the sole purpose of checking whether the application is usable, is functioning as expected or not, and is secure. The survey being completed by many users and bugs and issues being detected; it was required to patch the program before releasing it to all the users. The UAT process involved comprehensive testing scenarios to cover various usage patterns and edge cases,

ensuring a thorough evaluation of the application's performance. User feedback was meticulously analyzed to identify common pain points and areas for improvement, leading to targeted updates and optimizations. Post-UAT, a final round of testing confirmed the effectiveness of the patches, validating that the application met all usability, functionality, and security requirements before its official launch.

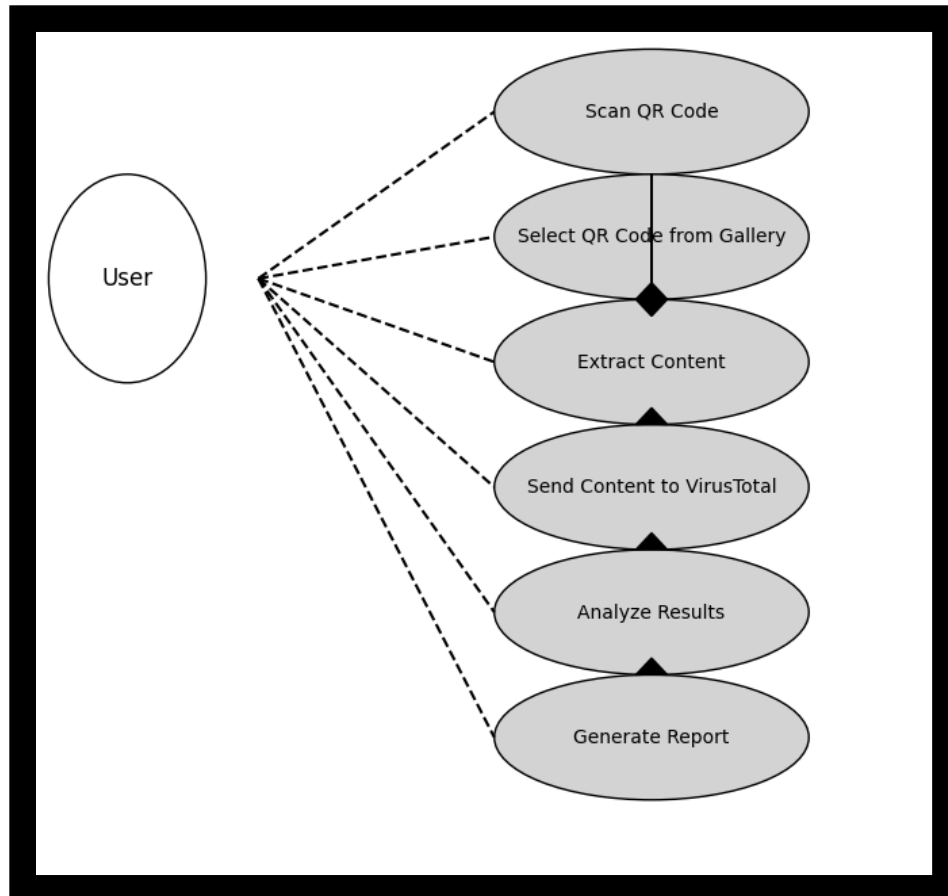


Figure 13: Use case Diagram of QR Shield Application

### 3.3.4 Deployment Phase

The application was prepared for production deployment. Building the highest level of automation in terms of CI/CD pipelines to allow the testing, as well as deployment to be a continuous process. We used the Android studio IDE to produce a signed APK file to release our application on the Google Play Store. Signing of the APK is a requisite step for distributing an application on the Google Play Store. Handling the task of uploading the application onto the Google Play App Store through this that using Google's guidelines are followed, and their submission processes.

## Chapter 4: Code Analysis & Evaluation

### 4.1 ScanningActivity Code Analysis

The ScanningActivity class covers the task of scanning QRs, the camera usage on the device as well as the URL analysis by using the VirusTotal API. It combines these features with another option to select a photo from the gallery represented by a photo icon. The code does definitely deal with the primary capability of it, but there is still space for better handling of errors, security, and code maintenance which in turn can help to increase its reliability, stability, and maintenance.

#### Key Components

##### 4.1.1 Imports

By coding, we have imported numerous Android and third-party libraries required for digital camera handling, image loading, networking, and UI components.

##### 4.1.2 Class Members

The functionalities of following class members are explained below:

###### 4.1.2.1 Camera Executor

Executor for camera operations.

###### 4.1.2.2 QR Scanner

QR scanning client from ML kit.

###### 4.1.2.3 Preview, title, urlTV etc

UI components like PreviewView, TextView, ImageView etc.

###### 4.1.2.4 Client

OkHttpClient for network requests.

###### 4.1.2.5 APIKey

This key is generated after signing up on VirusTotal and will be used to access all the functions of VirusTotal.

###### 4.1.2.6 PdDialog

Show ProgressDialog for showing progress while URL is evaluated.

#### 4.1.2.7 PickMedia

ActivityResultLauncher for media picking.

#### 4.1.2.8 ResponseStr

String to store reaction from the VirusTotal API.

### 4.1.3 onCreate() Method

This method initializes the UI components and sets up the camera or photo picker based totally at the scannerType passed thru purpose.

### 4.1.4 openPhotoPicker() Method

It uses the TedImagePicker library in allowing user to select a picture from the gallery. When an image is picked, it uses the imageView to put the photo into it and as soon as the Photo is uploaded, the analysis starts with a slight delay.

### 4.1.5 startCamera() Method

This method computes, sets up, and fire up the camera to scan QR code. It instantiates alive camera preview screen and the getAnalysis method using CameraX API.

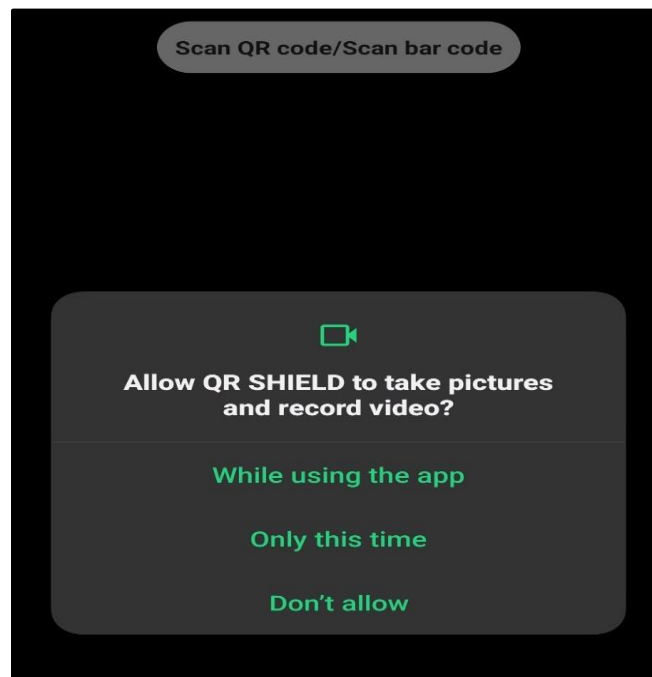


Figure 14: Application Permissions

#### 4.1.6 analyzeImage() Method

Processes the captured image or selected image URI into firebase to be compared to QR code with help of Machine Learning Kit. In a situation when a QR code is detected, then URL is displayed and its analysis is started.

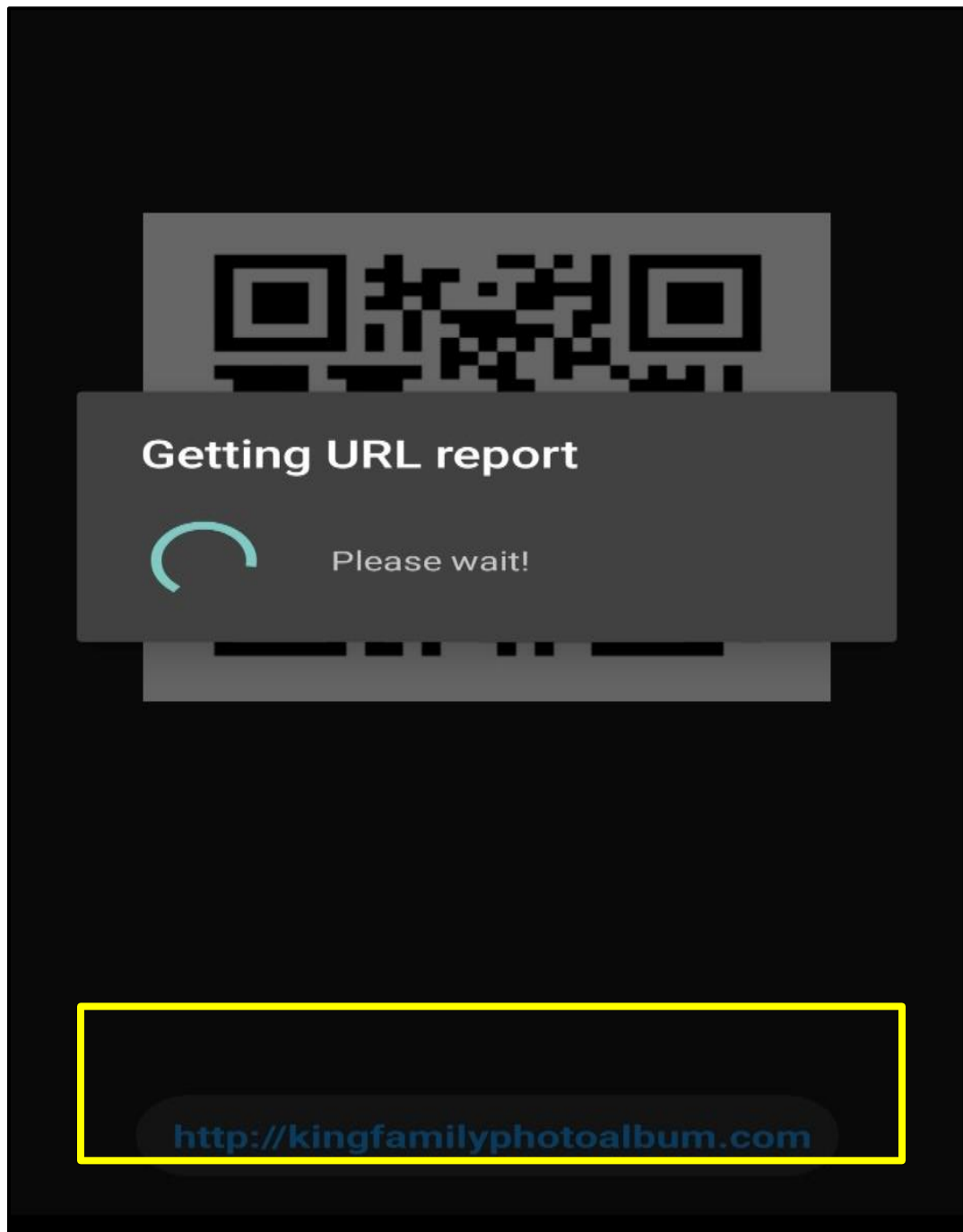
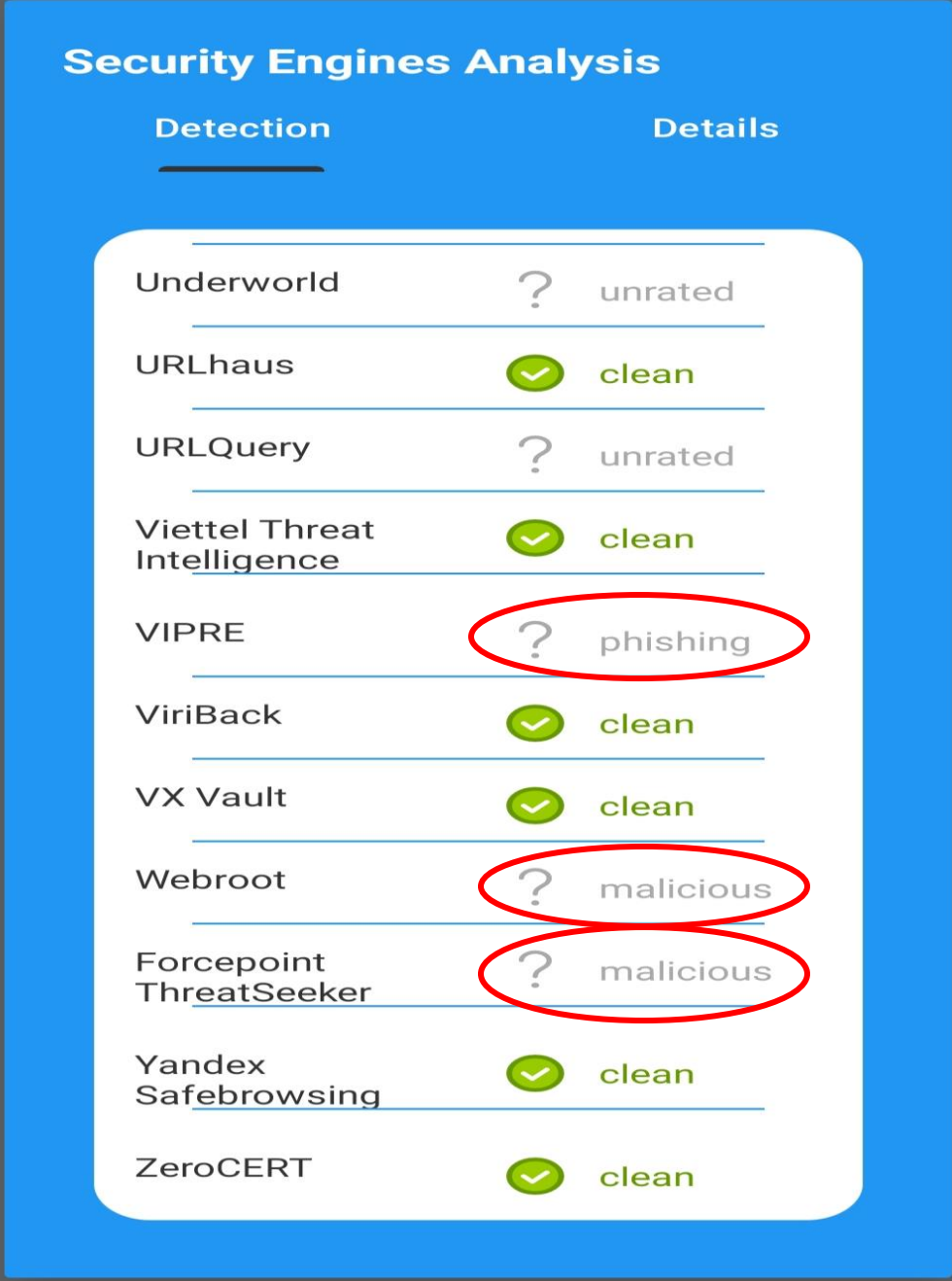


Figure 15: URL Extraction

#### 4.1.7 analyzeUrl() and getReport() Methods

It requests the VirusTotal API to perform URL analysis. It Displays a ProgressDialog during analysis the and stores the extracted analysis data in a database.



Detection	Details
Underworld	? unrated
URLhaus	✓ clean
URLQuery	? unrated
Viettel Threat Intelligence	✓ clean
VIPRE	? phishing
ViriBack	✓ clean
VX Vault	✓ clean
Webroot	? malicious
Forcepoint ThreatSeeker	? malicious
Yandex Safebrowsing	✓ clean
ZeroCERT	✓ clean

Figure 16: Detailed Analysis of URL

### **4.1.8 Permissions Handling**

Checks and requests camera permissions if not granted.

## **4.2 Evaluation of ScanningActivity Code**

### **4.2.1` API Key Exposure**

API keys should not be embedded directly into the code, otherwise they will not be secure. Using Android Keystore we have stored the critical data.

### **4.2.2 Error Handling**

The need for error handling during network requests, images loading, plus other important functions was implemented to ensure our user experience smoothness.

### **4.2.3 UI Feedback**

The application provides visual feedback to the user instance while scanning and analyzing the data, with the help of indicators or missing written messages.

### **4.2.4 Code Organization**

The code was split into several functions and methods with the aim of having it well-structured and easily maintainable. Classifying networking ideas into a different class so as to improve the separation of concerns principle.

### **4.2.5 Memory Management**

We paid a lot of attention to the proper memory allocation offered during dealing with image and camera resources, for avoiding the problems of memory leaks.

### **4.2.6 Optimizations**

This code optimizes image loading and processing for better performance. We, therefore, have fetched scanned URLs and results of the analysis in order to minimize the non-productive requests.

### **4.2.7 Security**

The application performs validation and auto-updating of URLs before an analysis is done so as to ensure secure process. An HTTPS protocol is used for network requests, which aims to prevent information leakage.



## **4.3 Result Activity Code Analysis**

### **4.3.1 Imports**

The code starts by importing necessary Android and third-party libraries. Imports need to be maintained which include things like UI components, intent handling, JSON parsing, and logging.

### **4.3.2 Class Declaration**

The ResultActivity class extends AppCompatActivity, which is a base class for activities that use the AppCompatActivity library features.

### **4.3.3 Properties**

Several private properties are declared at the class level to reference various UI components:

#### **4.3.3.1 url**

TextView to display the URL.

#### **4.3.3.2 img**

ImageView shows images representing secure, or insecure URLs.

#### **4.3.3.3 msg**

TextView to display a message about the URL's maliciousness.

#### **4.3.3.4 cont, abort, fullRepo**

TextViews representing clickable buttons or links in the UI.

### **4.3.4 onCreate Method**

The onCreate method is overridden to initialize the activity when it's created.

### **4.3.5 Initialization**

The setContentView method sets the activity's layout using the activity\_result layout resource. UI components are initialized using findViewById.

### **4.3.6 Retrieving Intent Extras**

The code retrieves two extras from the intent:

#### **4.3.6.1 data**

A URLResult object passed as a parcelable extra.

#### **4.3.6.2 response**

A string representing the response from some previous operation.

### **4.3.7 Impact of SDK Version**

Smartphones operating on Android differ in their ways of data retrieval which is determined by the Android SDK version. In SDK versions such as TIRAMISU or higher the expression goes with the intent. `getParcelableExtra`. Otherwise, it has 'intent' keyword built in. `getParcelableExtra`.

### **4.3.8 Updating UI Components**

#### **4.3.8.1 url.text**

Sets the text of the url TextView to the URL from result.

#### **4.3.8.2 img.setImageDrawable**

Inspects img ImageView aristocracy based on the malicious property of result.

#### **4.3.8.3 msg.text**

Sets the text of the msg TextView using a formatted string that includes malicious and total properties of result.

### **4.3.9 Button Click Listeners**

#### **4.3.9.1 cont**

Starts a WebviewActivity passing the URL as an extra when clicked.

#### **4.3.9.2 abort**

Navigates back to MainActivity and clears the activity stack.

#### **4.3.9.3 fullReport**

Shows an AnalysisFragment passing the response string as an argument when clicked.

## **4.4 Evaluation of Result Activity Code**

### **4.4.1 Null Safety**

Implementation of code assumes that the words in 'intent\_extras' are always defined and it never returns any null values. Safe calls or null checks are applied to avoid clearly defined NullPointerException.

### **4.4.2 Code Duplication**

To improve readability, the code can be refactored to a separate function for displaying the chosen image and malicious score. This makes the function more reusable and reduces code redundancy.

### **4.4.3 Intent Extras Keys**

Magic strings like "data" and "response" can be replaced with constants to avoid typos and increase code maintainability.

### **4.4.4 UI Responsiveness**

The UI update operations are done sequentially on the main thread. In case the application executes operation over and over again, it may lead to UI locking. Consider using cooperative coroutines for time-consuming tasks or spawn background threads.

### **4.4.5 Resource Management**

The AnalysisFragment instance created on button click might not be properly managed. Ensure to handle its lifecycle properly to avoid memory leaks.

### **4.4.6 Error Handling**

The absence of such corrections may bring about cases where the default falls into intent extras are missing or invalid. In this regard, we'll bring in error handling to manage situations like this in a neat manner.

In the evolving world of digital devices, applications in mobile devices play a crucial role in user's online security. QR Shield comes as a unique solution particularly designed to address the thriving security concerns posed by QR codes. This thesis presents the phases of development and integrated functions of QR Shield, a user-friendly mobile application that enables individuals to use the QR code with confidence in a secure environment.

The strength of QR-SHIELD lies in its multi-layered approach to QR code analysis. The application uses a very large database of malicious signatures which are regularly updated by leading antivirus engines that helps QR Shield to identify known threats effectively. These antivirus engines act as the main line of defense, countering attacks by malicious actors to compromise user devices and exploit weaknesses. QR Shield also considers the potential for attackers of slightly modifying links to evade traditional security measures (**i.e Signature Based Detection**). The application also applies behavioral analysis using the platform, VirusTotal through its API key thereby checking the behavior of the link and uncovering any activity that might signify malicious intent.

The application has a very user-friendly interface which plays a pivotal role in helping its users to make informed decisions. QR SHIELD shows a concise report to the users which helps them to understand the potential security risks associated with a particular scanned QR code. The report includes the number of antivirus engines that flagged the link as malicious and informs us about the category or type of threat a malicious QR Code is part of. Thereby using these granular level of details QR SHIELD provides its users the necessary knowledge to make well-considered decisions, and to continue having a safer online experience.

In conclusion, QR Shield adds a great contribution to the domain of mobile applications security. It bridges the gap between signature-based detection which is reactive and behavioral analysis that is proactive. Moreover, continuous integration of the latest and emerging threat intelligence and ongoing refinement of the behavioral analysis algorithms will further improve the QR Shield's effectiveness. As the landscape of cyber threats continues to evolve, QR Shield stands poised to remain at the forefront, safeguarding users from fraudulent and malicious QR codes, ultimately fostering a more secure and trusted digital experience.

## Chapter 6: Future Work

## **6.0. Future Steps for Commercializing QR Shield**

QR Shield has great potential to be a leading security application in the mobile landscape. However, to transform the potential of QR SHIELD into commercial success, many key steps need to be followed. This section outlines these future steps, focusing on important aspects for a successful and smooth commercialization journey.

### **6.1. Refining User Experience and Design:**

#### **6.1.1. Intuitive Onboarding:**

A user-friendly onboarding experience is paramount. Consider interactive tutorials, clear explanations of functionalities, and a visually appealing interface to ensure a smooth learning curve for users of varied technical backgrounds.

#### **6.1.2. Customization Options:**

Implementing features like customizable scan history filters, the ability to prioritize specific antivirus engines, and different levels of detail in reporting would cater to diverse user preferences.

#### **6.1.3. Multilingual Support:**

Expanding language options will broaden QR Shield's reach and cater to a global audience, enhancing accessibility.

### **6.2. Robust Testing and Security Audits:**

#### **6.2.1. Beta Testing:**

Conducting thorough beta testing with diverse user groups will identify potential usability issues and refine the application's overall functionality.

#### **6.2.1. Penetration Testing:**

Engaging recognized security firms to conduct penetration testing would uncover potential vulnerabilities and ensure the application's integrity before release.

#### **6.2.1. Compliance Certifications:**

Investigating relevant security and privacy compliance certifications (GDPR or SOC) would demonstrate QR Shield's commitment to user data protection and strengthen user trust.

### **6.3. Building a Scalable Infrastructure:**

#### **6.3.1. Server Load Management:**

As the user base grows, the application needs to handle increased scan requests. Planning for server infrastructure scalability will ensure smooth handling of peak traffic.

#### **6.3.2. API Integration Optimization:**

Analyzing VirusTotal API usage patterns and implementing optimizations, if necessary, will ensure efficient and cost-effective utilization of their services.

#### **6.3.3. Database Maintenance Strategy:**

Developing a robust strategy for database maintenance will be crucial for maintaining the integrity and efficiency of the comprehensive threat database.

### **6.4. Marketing and User Acquisition:**

#### **6.4.1. Targeted Marketing Campaigns:**

Crafting targeted marketing campaigns that highlight QR Shield's unique strengths and cater to specific user demographics like security-conscious individuals or frequent QR code users will be essential.

#### **6.4.2. App Store Optimization (ASO):**

Optimizing the application listing in relevant app stores to enhance search visibility will drive organic user acquisition.

#### **6.4.3. Strategic Partnerships:**

Exploring collaborations with smartphone manufacturers, antivirus software companies, or digital security awareness initiatives can amplify QR Shield's reach.

### **6.5. Monetization Strategy:**

### **6.5.1. Freemium Model:**

Consider offering a freemium model with basic functionalities available for free and premium features, like advanced reporting or unlimited scans, accessible through subscriptions.

### **6.5.2. In-App Advertising:**

Implementing non-intrusive, targeted in-app advertisements can be another revenue stream while maintaining user experience.

### **6.5.3. Strategic Integrations:**

Partnering with security service providers to offer QR Shield as an integrated service within their existing security packages would be a lucrative monetization avenue.

## **6.6. Continuous Development and Feature Enhancement:**

### **6.6.1. Machine Learning Integration:**

Having this served on an active set, i.e. implementation of smart machine learning algorithms, and making an adjustment to the detection process beforehand could let you to have it automatically detected.

### **6.6.2. QR Code Content Scanning:**

Such a development can be arrived at by putting in place a monitoring system that scans QR codes and their corresponding text often URLs. Besides this should the system check the e-mail address and numbers as well.

### **6.6.3. Offline Scanning capabilities (Optional):**

Having a switch to an offline assuming a tool that measures minimum functionality could also be considered another solution which is easier to operate.

## **6.7. Building a Strong Community and Support Network:**

### **6.7.1. User Feedback**

Establishing a robust feedback loop with users is essential for continuous improvement. Regular surveys, in-app feedback options, and user forums can provide valuable insights and help prioritize feature updates and bug fixes.

#### **6.7.2. Customer Support:**

Implementing a responsive customer support system, including live chat, email support, and comprehensive FAQs, will enhance user satisfaction and retention. Consider also offering premium support for paying users.

#### **6.7.3. User Community Engagement:**

Creating and nurturing an active user community through social media, online forums, and regular newsletters can foster a sense of belonging and loyalty among users. Encourage user-generated content and testimonials to build trust and credibility.

### **6.8. Expanding Functionality:**

#### **6.8.1. Integration with Other Security Tools:**

Allowing QR Shield to integrate with other popular security tools and antivirus software can provide users with a more comprehensive security solution and increase the application's value proposition.

#### **6.8.2. Enterprise Solutions:**

Developing versions of QR Shield tailored for enterprise use can open new revenue streams. Features like centralized management, bulk scanning capabilities, and integration with enterprise security frameworks can make QR Shield attractive to businesses.

#### **6.8.3. Emerging Technologies:**



Keeping an eye on emerging technologies like blockchain for enhanced security or augmented reality (AR) for intuitive QR code interactions can ensure QR Shield remains at the forefront of innovation in the security space.

## **6.9. Global Expansion Strategy:**

### **6.9.1. Localization:**

Beyond multilingual support, localization efforts including region-specific features, compliance with local data protection laws, and region-targeted marketing campaigns will be crucial for successful global expansion.

### **6.9.2. Regional Partnerships:**

Forming strategic partnerships with local tech companies, security firms, and mobile carriers can facilitate smoother entry into new markets and accelerate user adoption.

### **6.9.3. Cultural Adaptation:**

Understanding and adapting to cultural nuances in different regions can enhance user experience and acceptance. Tailoring the user interface, marketing messages, and customer support to align with local preferences and behaviors will be beneficial.

## **6.10. Data Privacy and Ethical Considerations:**

### **6.10.1. Transparent Data Practices:**

Clearly communicating privacy policies and giving users control over their data will build trust and comply with privacy regulations.

### **6.10.2. Ethical AI Use:**

When integrating machine learning and AI, it's important to ensure these technologies are used ethically, avoiding biases and ensuring fairness in the detection and analysis processes.

### **6.10.3. Regular Audits:**

Conducting regular audits of data practices and security measures will ensure ongoing compliance with legal standards and reinforce the application's commitment to protecting user data.

## **6.11. Performance Optimization:**

### **6.11.1. Speed Enhancements:**

Optimizing the application's performance to ensure fast scanning and minimal latency will improve user satisfaction and competitiveness in the market.

### **6.11.2. Battery Efficiency:**

Developing the application to be energy-efficient will enhance its appeal, especially among mobile users who are concerned about battery life.

### **6.11.3. Resource Management:**

Efficient management of device resources (CPU, memory) will ensure the application runs smoothly on a wide range of devices, from high-end smartphones to more basic models.

## **6.12. Post-Launch Monitoring and Updates:**

### **6.12.1. User Behavior Analytics:**

Analyzing user behavior and engagement metrics post-launch will provide insights into how the app is being used and identify areas for improvement.

### **6.12.2. Regular Updates:**

Commit to regular updates that not only address bugs and security vulnerabilities but also introduce new features and enhancements to keep the app competitive and engaging.

### **6.12.3. Long-term Roadmap:**

Developing a long-term product roadmap will help in strategic planning and ensure continuous development aligned with market trends and user needs.

## References and Work Cited

1. “The Evolution and Emergence of QR Codes” by Celalettin Aktaş
2. “QR codes and their applications for libraries: A case study from the University of Bath Library”
3. “Principles of Web API Design: Delivering Value with APIs and Microservices” by James Higginbotham
4. “Principles of Web API Design” by James Higginbotham
5. <https://www.cambridgescholars.com/resources/pdfs/978-1-4438-5065-0-sample.pdf>
6. <https://docs.virustotal.com/docs/api-overview>
7. <https://www.youtube.com/watch?v=5DEHmN4PmA0>
8. <https://learntodroid.com/how-to-create-a-qr-code-scanner-app-in-android/>
9. <https://bing.com/search?q=develop+QR+code+scanner+Android+tutorial>
10. <https://publicapis.io/virus-total-api>
11. Ahmed, T., Shafique, M. F., & Tian, Z. (2021). Machine learning based phishing detection for QR codes. *Security and Communication Networks*, 14(17), 5892-5903.
12. Alqahtani, S., & Liu, X. (2022). A review on security vulnerabilities of QR code applications. *Journal of Information Security Applications*, 70, 101823.
13. Chen, C., Xu, X., & Zhu, Y. (2020). Dynamic QR code: A novel approach to enhance security and user experience. *Security and Communication Networks*, 13(16), 3427-3437.
14. Chen, H., Zhu, F., Jiang, X., & Li, Y. (2021). QR code and its applications in logistics and supply chain management. *International Journal of Production Research*, 59(13), 3809-3824.

15. Denso Wave Inc. (2023). QR Code: History & Usage. Retrieved from <https://www.denso-wave.com/en/technology/vol1.html>
16. European Union Agency for Cybersecurity (ENISA). (2020). Mobile Malware: Threats and Mitigations. Retrieved from <https://www.enisa.europa.eu/publications/malware/@@download/fullReport>
17. Fu, X., Zhu, Y., Wu, L., & Sun, W. (2022). Enhancing QR code security with threat intelligence platform. *IEEE Access*, 10, 17476-17488.
18. Gupta, M., Jain, P., & Singh, P. K. (2021). Security vulnerabilities and countermeasures in QR code mobile applications. *International Journal of Network Security & Its Applications (IJNSA)*, 13(4), 239-248.
19. Johnson, R. D., & Hui, H. K. (2019). QR codes in tourism: A literature review and future research directions. *Journal of Travel Research*, 58(1), 124-138.
20. Kim, Y., Park, J., & Lee, S. (2020). A review of user experience studies on QR code applications. *Journal of Information Processing Systems*, 16(2), 220-232.
21. Li, T., & Gong, M. (2021). A review on internet privacy and security. *Journal of Network and Computer Applications*, 182, 103032.
22. Li, X., Wang, Y., & Chang, K. (2020). A review of QR code applications in the cultural heritage domain. *Multimedia Tools and Applications*, 79(13-14), 9277-9302.
23. Oh, H., Park, S., & Kim, H. C. (2020). A review of QR code security: Current threats and countermeasures. *Sensors*, 20(15), 4342.
24. Zhang, Y., Zhu, Y., Wu, L., & Li, H. (2019). A review on attacks and defenses on QR codes. *ACM Computing Reviews (CSUR)*, 52(5), 1-37.