بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

SUDAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

COLLEGE OF GRADUATE STUDIES

A Secure Android SMS Application Using AES

تطبيق اندرويد آمن للرسائل القصيرة باستخدام خوارزمية AES

A Thesis Submitted In Partial Fulfilment for the Requirement of the Degree of Master in Computer Science

Prepared by:

Sara Hamad Rahama Heemadan

Supervised by:

Dr. Faisal Mohammed Abdallah

November .2020


بسم الله الرحمن الرحيم

الآيـــة:

قَالَ تَعَالَى: ﴿ أَلَمْ نَشْرَحْ لَكَ صَدْرَكَ ۝١ وَوَضَعْنَا عَنكَ وِزْرَكَ ۝٢ ٱلَّذِىٓ

أَنقَضَ ظَهْرَكَ ۝٣ وَرَفَعْنَا لَكَ ذِكْرَكَ ۝٤ فَإِنَّ مَعَ ٱلْعُسْرِ يُسْرًا ۝٥ إِنَّ مَعَ ٱلْعُسْرِ يُسْرًا ۝٦

فَإِذَا فَرَغْتَ فَٱنصَبْ ۝٧ وَإِلَىٰ رَبِّكَ فَٱرْغَب ۝٨ ﴾ صَدَقَ اللهُ العَظِيمْ

سورة الشرح الآية من (1-8)

ACKNOWLEGEMENT

First and foremost, thank you God to give me another chance to complete my higher education and to learn enormous skills and knowledge through all these years of study at the university. I would like to express my sincere gratitude to Dr. FAISAL M ABDALLAH for his patience and guidelines through this project His guide lines helped me to comprehend how to formulate my ideas and solutions into a structured design and acceptable work.

I am also very grateful to all the people who helped me to finalize this work, for all of their sincere cooperation, brainstorming and contribution.

God bless you all.

ABSTRACT

Short Message Service (SMS) is a text messaging service component of mobile communication systems, It uses standardized communications protocols to exchange short text between mobile devices. SMS does not have any built-in procedure to offer security for the text transmitted as data. Most of the applications for mobile devices are designed and developed without taking security into consideration. In Practical use, SMS messages are not encrypted by default during transmission. Therefore, this subsequent project designs and implements a secure SMS android application. Cryptographic manipulation of the data is performed using AES algorithm to secure the data. To increase security and data privacy the application generates a random key using AES algorithm and the key is saved after converting it to QR code format, which is essential for the safe transmission of confidential data over the GSM network.

الملخـــص

خدمة الرسائل القصيرة (SMS) هى احد مكونات خدمة الرسائل النصية لأنظمة الاتصالات المتنقلة، وتستخدم برتكولات اتصالات موحدة لتبادل النصوص القصيرة بين الاجهزة المحمولة .لا تحتوى الرسائل القصيرة على أى إجراءات مضمنة لتوفير الأمان للنص المرسل كبيانات. تم تصميم معظم تطبيقات الأجهزة المحمولة وتطويرها دون مراعاة الأمان فى الاستخدام كما لا يتم تشفير الرسائل افتراضيا اثناء الانتقال والتخذين. فى هذا البحث تم تصميم وتطوير تطبيق اندرويد يقوم بتشفير الرسائل النصية باستخدام خوارزمية AES والتى تقوم بتوليد مفتاح عشوائى ومن ثم تحويله إلى صورة QR لضمان نقله بسرية وأمان عبر شبكة GSM.

TABLE OF CONTENTS

CONTENT	PAGE NO
Acknowledgement	I
Abstract	II
Abstract (Arabic)	III
List of Contents	IV
Chapter one: Introduction	
1.1 Introduction	1
1.2 Problems Statement	2
1.3 Research Objective	2
1.4 Research Solution	2
1.5Research Importance	3
1.6Research Scope	3
1.7Research Methodology	3
1.8Software Requirements	4
1.9Hardware Requirements	4
1.10Research presumption	4
1.11Expected Results (Hypotheses)	5
1.12Research Layout	5
Chapter Two: History and Background and Literature Review	
2.1 Security services for Mobile Instant Messaging	6
2.1.1 Confidentiality	6
2.1.2 Authentication	6
2.1.3 Integrity	7

2.2 Security Challenges for Mobile Device Users 7
2.2.1 Insecure Data Storage 7
2.2.2 Physical Security 7
2.2.3 Mobile Browsing 8
2.2.4 Multiple User logging 8
2.2.5 Client Sid Injection 8
2.2.6 Improper Session Handling 8
2.2.7 Weak Authentication and Brute Force Attack 9
2.3 Mobile Threats and Vulnerabilities 9
2.3.1 Mobile Threats 9
2.3.2 Privacy Threats 9
2.3.3 Network Based Threats 11
2.3.4 Web Based Threats 11
2.3.5 Mobile Vulnerabilities 12
2.3.6 Defensive Mechanisms 12
2.4 Cryptography 13
2.5 Advance Encryption Standards Algorithm 14
2.5.1 Following are the four steps 14
2.5.2 Decryption of Algorithm 14
2.5.3 Implementation and Pseudo Code of Application 15
2.5.4 AES parameters 16
2.5.5 Security 16
2.6 Digital Signature 17
2.6.1 How Digital Signatures Work 17
2.6.2 How To Create Digital Signature 17
2.6.3 Classes of Digital Signatures 18

2.6.4 Digital Signature VS Electronic Signatures 18

2.7 Android 19

2.8 Android Architecture 19

2.9 Previous study 20

2.10 User Interface Design Principles 21

2.11 Mobile Interface Design Principles 24

Chapter Three Methodology and Analysis and Requirement

3.1 Introduction 28

3.2 Flow chart 28

3.2.1 Transmitter end 28

3.2.2 Receiver end 28

3.2.3 the device sanding the text message 29

3.2.4 the device receiving the text message 30

3.3 Sequence Diagram 33

3.4 Software Requirement 32

3.5 Hardware Requirement 35

3.6 User Requirement 35

Chapter Four: Design and Implementation and Results

4.1 introduction 37

4.2 User Interface for Mobile Application 37

4.3 User Interface for Registration and Setting 43

4.4 Results of Sender's 46

4.5 Results of Receiver 55

4.6 Reviewing Objectives 60

4.7 Research Limitations 60

4.8 Analysis and Results 60

Chapter Five : Conclusions and Recommendations

5.1Conclusions	62
5.2 Recommendations	62
References	63
Appendix	65

Chapter One

Introduction

CAPTER 1

INTRODUCTION

1.1 Introduction:

The age of information technology has transformed the ways human communicate with each other, introducing emails, messaging, using social networking sites and many more. The uses of computers, smart phones and clouds has become an integral part of our life for sharing information, but SMS is one of the popular ways of communication. Sending of a message is very direct and one of the easiest ways to send text. [1]

The communication via messages is preferred for privacy reasons. Conversation voice could be heard by others and for practical reasons and noise could hinder the conversation, many people choose to use text messages as are cheaper than voice call. Security issues such as authenticity and confidentiality of data or information still cannot be guaranteed. It is difficult to protect the massage during transmission when the user sends confidential data. Of all these means have also encouraged the misuse of information by third parties which can steal private information consequently, the security of SMS is an important issue as no secure medium is provided.

Based on the above, came the idea of using encryption to provide end-to-end security for SMS. This research implements an Android application using AES algorithms and digital signature to enhance the confidentiality, integrity, and authentication of data. The application encrypts the text message before it is sent and decryption is done at the recipient. This way the message will be unreadable during transmission.

1.2 Problem statement:

Nowadays, many people want to communicate with each other, use different methods for this, including SMS. This service is not very safe in the current environment. When sending text messages via a mobile phone, they can be intercepted on their way to the recipient, this affects the confidentiality, integrity and authenticity of the messages.

Therefore, it is necessary to provide a technique to preserve the messages even if exposed to any attack or malicious intents it is found encrypted and therefore meaningless.

1.3 Research objectives:

The aim of this research is to exploit the features of the mobile device and make the most of them to secure short text messages, taking into account the link between mobile features and modern technologies used in data preservation and unauthorized access, this can be summarized as follows:

I. The study explains why it is very necessary to ensure that even when an intruder or unauthorized user successfully obtains access to some message the confidentiality and integrity of the information remain uncompromised.
II. The study helps to choose the correct solution verify the integrity and authenticity of the message sent on mobile applications.
III. Reduce the risk message mobile so that no other person except the intended recipient can read.
IV. Guarantee the main goals of cryptography, data confidentiality and data integrity and guarantee end to end security.

Verify the integrity and authenticity of messages sent via SMS

Ensure that no person other than the recipient can read the message even under unauthorized access.

1.4 Proposed solution:

This research will improve the SMS sent via the smart phone, which will increase the security and prevent the attacker from getting any useful information even if receives the text message, by designing a distinct and effective Android application.

1.5 Research importance:

The Three security goals are confidentiality, integrity and availability. All information security measures try to address at least one of the three goals, but this study attempts to apply two security concepts (protect the confidentiality and preserve the integrity of data), to obtain the highest degree of security and integrity of transmitted data. SMS (confidentiality, integrity) are selected to ensures the safety of the contents of the message by preventing hackers that are efficient in breaching the security and protect the information from illegal use and unauthorized access. So, the text messages, which are sent and stored on servers or computers in plain text, can be protected from interception during transit.

1.6 Research scope:

The scope of the search is limited to encrypting a text message, then sending it using the SMS system, ends when decrypting it at the receiver. AES algorithm is used to encrypt messages. The application is used to on Android smart phones.

1.7 Research methodology:

This research relies on a methodology based on making the most of the security features provided by the libraries of the Android operating system and optimizing the use of modern technologies to reach a safe way to send short text messages over networks and in order to achieve this:

Initially, a random encryption key is generated depending on the username and password. Then, it makes use of the AES encryption algorithm, built into the Android system libraries, to encrypt the text message. And then the applied of digital signature technology between the sender and receiver applications of the short text message. Finally, the encryption key is shared in the form of a QR code.

1.8 Software requirements:

Following are the software requirements to develop the Android application:

1) Windows OS (8+).
2) Java virtual machine JVM.
3) Kotlin virtual machine KVM.
4) Android Development Tools ADT.
5) Android Studio IDE for Android Application.

1.9 Hardware requirements:

To ensure the efficiency, activation of all the advantages of the Android system and make the most of them the following requirements are recommended:

1) Processor – i5.
2) Memory – 6GB.
3) Hard Disk –320/500 GB.
4) Two SIMs support the short text message service
5) Two Android devices for running and testing the application.

1.10 Research presumption:

The following presumptions are made:

1) The intruder cannot enter into the system even if they obtained the username and password.
2) Mobile based authentication ensures more strong security than the traditional ways.
3) The system user should not think too much about generating and saving long, complex passwords for more security.

1.11 Expected results:

This research seeks to:

1) Increase the security and reliability of short text messages with the block encryption algorithm AES on the Android application.
2) Short text message encryption application to work on the mobile phone without the need for additional encryption devices.
3) Convenient and easy to implement on mobile devices using the AES encryption algorithm and digital signature with a QR code.

1.12 Research layout:

- ✓ ***Chapter 1. Introduction*:**

Consists of the introduction, the problem, objectives, proposed solution, importance, scope, methodology, software and wore requirements, presumption and finally expected results.

- ✓ ***Chapter 2. Background and literature review:***

Contains the background and discusses the literature review and previous researches, studies and publications done in the same area, Moves onto the principles of designing user interfaces and mobile phone.

- ✓ ***Chapter 3. Research methodology:***

Discusses the methodology that will be followed in this research analysis software, hardware requirements used in the construction and development.

- ✓ ***Chapter 4. Design and implementation and research results:***

In addition to the design phase of the research, and discuss the results obtained after testing and checking the application of mobile phone use in generating secure authenticated random password.

- ✓ ***Chapter 5. Conclusion and recommendations:***

This is the last chapter, which includes the findings, planning for future works and recommendations that improve application performance.

Chapter Two

Background

CAPTER 2

BACKGROUND AND LITERATURE REVIEW

2.1 Security services for mobile instant messaging:

In order to evaluate any chat application or services for mobile instant messaging from the security point of view, relevant threats to such application should be identified and described. In the following sections brief descriptions about different security aspects are explained. Security has three key aspects: confidentiality, integrity and availability. [16]

2.1.1 Confidentiality:

Confidentiality means messages which are exchanged by two parties through a communication channel should be readable only to the intended parties. In order to achieve such a goal, encryption is the mechanism that provides confidentiality between two parties. A message is encrypted by a cryptographic technique and this encrypted message can only be readable by the intended party. [16]

2.1.2 Authentication:

Authentication is one of the most important aspects of security, where an entity should identify itself before or during the communication. This avoids any type of attack or malicious activity by which a malicious user impersonates the user and identifies himself as the real user to the server. There are two types of authentication schemes known as weak authentication and strong authentication. [16]

2.1.2.1 Weak authentication:

One factor authentication means that the entity uses only one type of identity credential such as a PIN or password-based authentication.

2.1.2.2 Strong authentication:

Usage of typically challenge-response cryptography. In this scheme the client needs to prove his identity and verify himself to the server with multiple factors. There are different practices to perform such authentication such as:

I. One-time passwords OTP the system issues one-time passwords based on this shared secret key.

II. Certificate-based authentication CBA is using asymmetric cryptography which provides public-private key cryptography.

2.1.3 Integrity:

Integrity ensures that a message has not been edited or changed during the transfer of it between entities. An attacker can eaves drop the communication channel and modify the message or even replace the message with a new one. Hashing is a mechanism to achieve such a goal in the world of information security. A cryptographic hash function is a function which maps an encrypted message to a fixed size length integer. A hash function is one-way function, meaning that if somebody has an output of a hash, it cannot be reversed. [16]

2.2 Security challenges for mobile device users:

Mobile device applications offer a level of convenience that the world never before considered, at any location (home, office, hotel, playground, road, parking, museum, travelling in different countries, or anyplace in the world), any mobile user can use applications to fulfil their daily needs, including communicating, buying, searching, making payments, selling, entertainment, and finding general information.

This extreme level of comfort has brought with it an extreme number of security risks, some of the mobile device challenges are described below, including ways that vulnerabilities and attackers are reducing mobile application freedom. [12]

2.2.1 Insecure data storage:

A user can suffer a data loss after losing a mobile device or experiencing interruption by some malicious application that deletes a user's most valuable information. In this way, all users are at risk by engaging in this type of activity, some common pieces of data are stored at high risk, including personal information, work information. [12]

2.2.2 Physical security:

Physically securing a mobile device is difficult, but when a mobile user is constantly using their mobile device (24×7×365) and it is lost, then the task becomes seemingly impossible, obviously, physical security is the greatest.

Concern for risk-free mobile devices if a person's mobile device is lost or stolen, the user's sensitive data may be misused by a thief, including personnel information, unsecured documents, business data, and files. [12]

2.2.3 Mobile browsing:

Mobile browsing is the best feature of any mobile device for providing the best use of internet applications. However, normally in mobile devices, a user cannot see the entire URL or web address, making it difficult to determine whether the web address or URL is safe. Thus, browsing can be used as a phishing related attack. [12]

2.2.4 Multiple user logging:

The execution of malicious programs on mobile devices over the internet occurs by application or web browsing client-side injection. Html injection, SQL injection, or another newer attack (abusing phone dialer, SMS).

Involves client-side injection. Hackers could load a text-based attack and exploit a targeted examiner. In this way, any data source can be injected, including resource targeted files or applications. [12]

2.2.5 Client side injection:

The execution of malicious programs on mobile devices over the internet occurs by application or web browsing client side injection. Html injection, SQL injection, or another newer attack (abusing phone dialer, SMS)

Involves client side injection. Hackers could load a text-based attack and exploit a targeted examiner. In this way, any data source can be injected, including resource targeted files or applications. [12]

2.2.6 Improper session handling:

For mobile devices, session handling is an identified security concern for web applications. Improper session handling has vulnerabilities that are pretty common when using internet applications over any platform like mobile devices or PCs, sessions with long

expiry times invite vulnerabilities when performing financial tasks, poor session management can provide clues to unauthorized access through session hijacking in mobile devices. [12]

2.2.7 Weak authentication and brute force attack:

Today, many applications rely on password-based authentication, as a single factor, the owners of these applications do not enforce strong passwords and the securing of valuable credentials, thus, users expose themselves to a host of threats, including stolen credentials and automated brute force attacks. [12]

2.3 Mobile threats and vulnerabilities:

A comprehensive overview of threats and vulnerabilities shows that cyber criminals are now focusing increasingly on mobile devices. Mobile devices use many useful applications on the internet, which makes them a prime target for attackers to destroy security mechanisms and cause threats, spread vulnerabilities, the distance between a hacker's capability and an organization's protection is widening day by day.

This tendency underlines the need for additional mobile device security cognizance, as well as more flexible, better integrated mobile device security solutions and policies, some significant mobile threats and vulnerabilities are described: [12]

2.3.1 Mobile threats:

Threats and attacks that worked well on personal computers are now being tested on unsuspecting mobile devices to see what works (mechanism) and, with protection increasing; there is an adequate number of easy targets.

Attackers are definitely penetrating the weakest point in the chain and improving on the most successful scams. Mobile attacks are basically divided into four categories in terms of user perspective, service/content provider Perspective, and network perspective, as listed below. [12]

2.3.1.1 Physical threats:

Mobile devices are designed to be used in daily life, and physical security is an important issue some of the physical threats are described below.

1. Bluetooth:

This is a short-range radio technology that provides wireless connectivity in very short ranges, and many potential threats, vulnerabilities, and exploits have been recognized with Bluetooth 16, malicious data are transferred to the other device by the Bluetooth services.

2. Lost or stolen mobile devices:

The loss or theft of valuable mobile devices is also a serious threat because these valuable applications and hardware devices can be resold on the market.

2.3.1.2 Application-based threats:

Many downloadable applications are available over the internet, and most of these have multiple security problems, can be classified as one or more of the following mobile applications.

1. Spyware: This is designed to collect personal, private data without a user's knowledge or endorsement.

2. Malware: Malicious software accomplishes malicious action after being installed in a user's mobile device without the user's knowledge or approval.

2.3.1.3 Vulnerable application:

Vulnerable applications are those applications that contain faults that can be exploited with malicious intent. They give an attacker permission to perform unwanted actions, access sensitive personal or business information, stop correctly performing activities, and download applications without approval.

2.3.2 Privacy threats:

Privacy threats can be caused by mobile applications in addition to malicious applications, for example the global positioning system GPS can provide information about any place a user visit.

An attacker or hacker can steal a user's information and identity, which can cause serious problems. [12]

2.3.3 Network-based threats:

Mobile devices provide the best support to cellular networks, as well as wireless LAN IEEE 802.11, both of which have different types of threats for the user; some network-based threats are described below. [12]

I. **Denial of service attack DOS:**

Means an attacker or hacker denies access to application services or other services.

II. **Network exploits:**

This type exploits the faults in the mobile device operating system or other application software that operates on a wireless or cellular network, when mobile devices are connected through a network.

III. **Mobile network services:**

Like MMS, SMS, and voice calls can also be used for attacking mobile devices.

IV. **Wi-Fi sniffing:**

Means intercepting data between the mobile devices and Wi-Fi access point from the air.

2.3.4 Web-based threats:

In mobile devices, there are always mobile users that use web-based applications over the internet. Thus, threats related to such activity is a major concern, and some researchers have proved that web-based threats are a much more serious problem for mobile devices, some web-based threats are described below. [12]

I. Drive by downloads:

This is a concept involving the automatic download of an application when visiting a web page (malicious web address).

II. Browser exploits:

This type of attack benefits from the vulnerabilities of a user's mobile web browser or an application (software) launched by the browser, such as PDF reader, flash player, and image viewer.

III. **Phishing scams:**

A means of obtaining sensitive or business information from a user by representing oneself as a reliable unit using a link on a social networking website, text message or email (spam) on a malicious website.

2.3.5 Mobile vulnerabilities:

Mobile vulnerability is a security expose that results from a mobile device weakness that the application developer for a mobile device did not expect to introduce and will fix once when it is discovered, Vulnerability includes three steps, a device has susceptibility, attackers access the flaw and a capable attacker exploits it. [12]

I. **Root kits:** Attain their malicious target by infecting the operating system.
II. **Worm:** Program code that makes multiple copies of itself from one mobile device to another, using diverse carriage techniques by the network.
III. **Trojan horse:** Installs other malicious (worm or botnet) applications and gathers sensitive information from the mobile devices.
IV. **Botnet:** A collection of compromised devices that are infected by virus programs that give an attacker the capability of remotely supervising them.

2.3.6 Defensive mechanisms:

Security and data privacy need multiple means of protection and restrictions on mobile devices. In general, the world now faces threats without correlation among all aspects of security breaches. In this regard, it is essential to ensure appropriate (restrictions and precautions) security mechanisms at all stages (development to final stage) when manufacturing mobile devices.

The section also takes into account, the precautions and actions from european advisory body opinions 34 regarding smart devices, these involve correlations among the application stores, operating system device manufacturers, application developers, non-recommended applications, and biometric approaches (authentication, verifications) in mobile devices to reduce security issues, data privacy threats, provide a secure mobile ecosystem. [12]

2.4 Cryptography:

Cryptography is the science of using mathematics to encrypt and decrypt data, it enables you to store sensitive information or transmit it across insecure networks like the Internet so that it cannot be read by anyone except the intended recipient.

Encryption is the process of converting ordinary information (called plaintext) into unintelligible gibberish (called cipher text).

Decryption is the reverse, in other words, moving from the unintelligible cipher text back to plain text; cryptographic algorithms can be divided into: [13]

2.4.1 Symmetric key algorithms:

Have the property that same secret keys are used for encryption and decryption. It is also called as private key algorithms; there are two types of symmetric-key algorithm: [13]

a. Block cipher:

In a block cipher, encryption and decryption operate on the basis of a block of symbols of particular size.

b. Stream cipher:

In a stream cipher, encryption and decryption operate on the basis of one symbol (a bit or byte) at a time.

2.4.2 Asymmetric key algorithms:

Use two different keys, public key for encryption and private key for decryption.

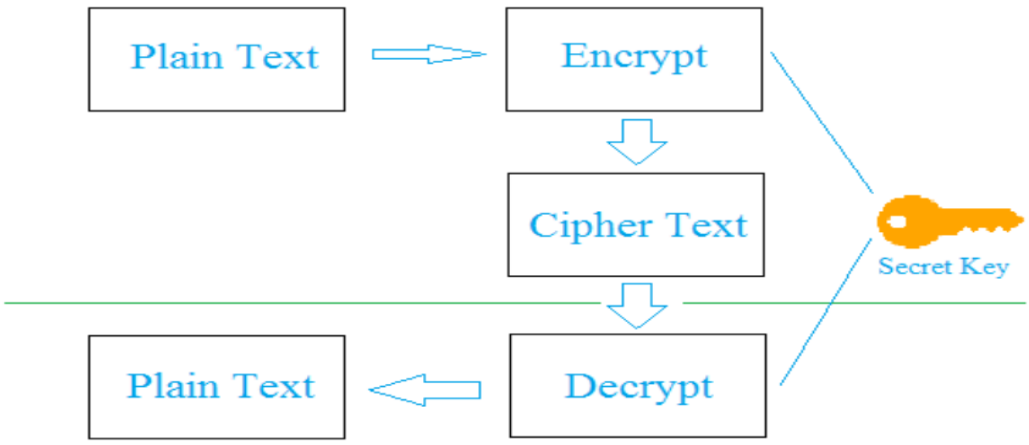

Figure (2.1): *Symmetric key algorithms.* [13]

2.5 Advance encryption standards algorithm:

The advanced encryption standard comprises three block ciphers, AES-128, AES-192 and AES-256. AES has a fixed block size of 128 bits and a key size of 128, 192, or 256 bits. The block-size has a maximum of 256 bits, but the key-size has no theoretical maximum. The cipher uses number of encryption rounds which converts plain text to cipher text. The output of each round is the input to the next round. The output of the final round is the encrypted plain text known as cipher text. The input given by the user is entered in a matrix known as state matrix. [13]

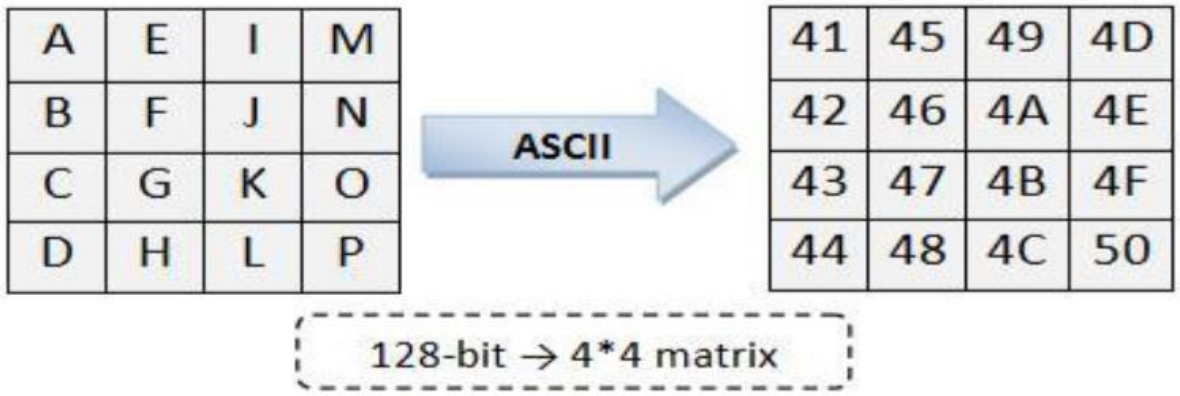

Figure (2.2): *State matrix.* [13]

2.5.1 Following are the four steps:

1. Sub bytes step:

This step is same as sub bytes step of AES algorithm. In the S-Box substitution step, each byte in the matrix is reorganized using an 8-bit substitution box; this substitution box is called the Rijndael S-Box. This operation provides the non-linearity in the cipher. The S-Box used is derived from the multiplicative inverse over GF 28, known to have good non-linearity properties, to avoid attacks based on simple algebraic properties, the S-Box is constructed by combining the inverse function with an invertible affine transformation. The S-Box is also chosen to avoid any fixed points (and so is a derangement), and also any opposite fixed points, this step causes confusion of data in the matrix. S-Box Substitution is carried out separately for LPT and RPT; this is the first step of iterative round transformation; the output of this round is given to the next round. [13]

2. Shift rows step:

The shift rows step is performed on the rows of the state matrix. It cyclically shifts the bytes in each row by a certain offset. The first row remains unchanged; each byte of the second row is shifted one position to the left. Similarly, the third and fourth rows are shifted by two positions and three positions respectively, the shifting pattern for block of size 128 bits and 192 bits is same.

3. Mix columns step:

In the mix columns step, the four bytes of each column of the state matrix are combined using an invertible linear transformation, a randomly generated polynomial is arranged in a 4*4 matrix, and the same polynomial is used during decryption. Each column of the state matrix is XOR-end with the corresponding column of the polynomial matrix. The result is updated in the same column; the output matrix is the input to add round key.

4. Add round key:

A round key is generated by performing various operations on the cipher key. This round key is XOR-end with each byte of the state matrix. For every round a new round key is generated using Rijndael's key scheduling algorithm.

2.5.2 Decryption of algorithm:

The encryption algorithm is referred to as the cipher and the decryption algorithm as the inverse cipher. In addition, the cipher and the inverse cipher operations must be executed in such a way that they cancel each other, the rounds keys must also be used in reverse order. The cipher text which is formed of 256-bit 4*8 matrix is the input for the decryption process. [13]

2.5.3 Implementation and pseudo code of application:

The algorithm can be implemented in any language, this algorithm can also be used in Image Processing .it has implemented it in java, java being an open source and platform independent language. [13]

2.5.4 AES parameters:

AES algorithm depending from the encryption mode whether it is Electronic Code Book ECB or Cipher-Block Chaining CBC takes two or three parameters. When encryption mode is ECB it takes two parameters a 128- bit plain text and 128, 192 or 256 -bit key. On

the other hand, as presented in Fig under, when encryption mode is CBC another extra parameter is needed it is a 128- bit initial vector; the output is always a 128 -bit. [13]

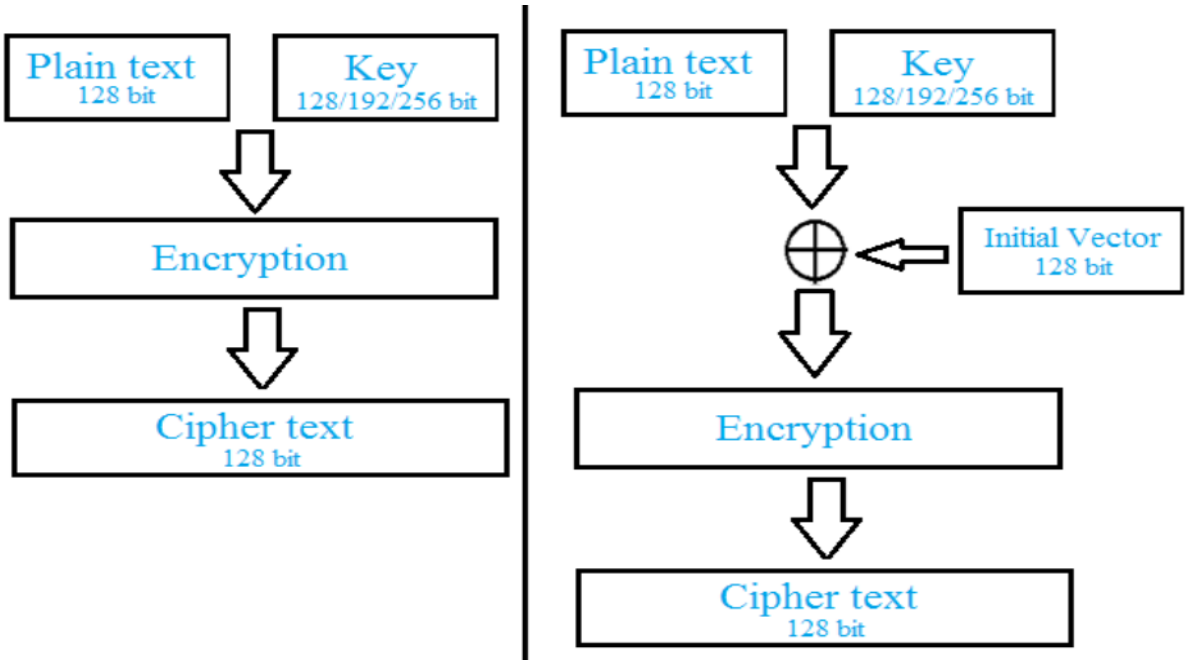

Figure (2.3): *AES encryption in ECB and CBC mode.* [13]

2.5.5 Security:

AES is considered to be the most secure symmetric algorithm, there are only a few attacks that can break it and they are applicable only on special conditions. To break it with brute-force attack it is almost impossible because there are too many combinations and it will take billions of years. Other attacks work only on special conditions with modified AES, but with the original one each of these attacks requires a lot of time and therefore it is not useful. In summary, AES is considered by NSA to be enough secure to protect even top-secret information, this information explains the best it's security level. [13]

Table (2.1): *Time needed to break AES in CBC mode via brute force attack.*

Key Length	Time needed to Break
128	1.2 * 1025 Years
192	2.4 * 1044 Years
256	5.1 * 1063 Years

2.6 Digital signature:

A digital signature is a mathematical technique used to validate the authenticity and integrity of a message, software or digital document, as the digital equivalent of a handwritten signature or stamped seal, a digital signature offers far more inherent security, and it is intended to solve the problem of tampering and impersonation in digital communications. [14]

2.6.1 How digital signatures work:

Digital signatures are based on public key cryptography, also known as asymmetric cryptography. Using a public key algorithm, such as RSA, one can generate two keys that are mathematically linked: one private and one public. For more on digital signatures work because public key cryptography depends on two mutually authenticating cryptographic keys. The individual who is creating the digital signature uses their own private key to encrypt signature-related data; the only way to decrypt that data is with the signer's public key, this is how digital signatures are authenticated. Digital signature technology requires all the parties to trust that the individual creating the signature has been able to keep their own private key secret. [14]

2.6.2 How to create a digital signature:

To create a digital signature, signing software such as an email program creates a one-way hash of the electronic data to be signed. The private key is then used to encrypt the hash, the encrypted hash along with other information, such as the hashing algorithm is the digital signature, the value of a hash is unique to the hashed data, any change in the data, even a change in a single character, will result in a different value, this attribute enables others to validate the integrity of the data by using the signer's public key to decrypt the hash.

If the decrypted hash matches a second computed hash of the same data, it proves that the data hasn't changed since it was signed. If the two hashes don't match, the data has either been tampered with in some way. A digital signature can be used with any kind of message whether it is encrypted or not simply so the receiver can be sure of the sender's identity and that the message arrived intact. Digital signatures make it difficult for the signer to deny having signed something assuming their private key has not been compromised as the digital signature is unique to both the document and the signer and it binds them together. This property is called non repudiation. Digital signatures are not to be confused with a digital

certificate; this is an electronic document that contains the digital signature of the issuing certificate authority. [14]

2.6.3 Classes of digital signatures:

Class 1: Cannot be used for legal business documents as they are validated based only on an email ID and username, and provide a basic level of security and are used in environments with a low risk of data compromise.

Class 2: Often used for e-filing of tax documents, including income tax returns and goods and services tax GST returns. Authenticate a signee's identity against a pr verified database, so are used in environments where the risks and consequences of data compromise are moderate.

Class 3: The highest level of digital signatures, class 3 signatures requires a person or organization to present in front of a certifying authority to prove their identity before signing. Are used for e-auctions, e-tendering, e-ticketing, court filings and in other environments where threats to data or the consequences of a security failure are high. [14]

2.6.4 Digital signature vs. electronic signature:

While digital signature is a technical term, defining the result of a cryptographic process that can be used to authenticate a sequence of data, the term electronic signature or e-signature is a legal term that is defined legislatively. For example, in the United States, the term was defined in the electronic signatures in global and national commerce act, passed in 2000, as meaning "an electronic sound, symbol, or process, attached to or logically associated with a contract or other record and executed or adopted by a person with the intent to sign the record, This means that a digital signature which can be expressed digitally in electronic form and associated with the representation of a record can be a type of electronic signature. [14]

2.7 Android:

Android is mobile operating system based on the Linux kernel and currently developed by Google. Android is popular with technology companies which require ready-made, low cost customizable operating system for high-tech devices. It is the customizable, easy to use operating system. Mobile phone users desire more secure and private communication in their

daily usage of their mobiles. This is especially important in communications of secret nature such as that in military and governmental communication. [15]

2.8 Android architecture:

The Android software stack consists of apps at the top, a Linux kernel with various drivers at the bottom, and middleware (an application framework, libraries, and the Android runtime) in the centre. [15]

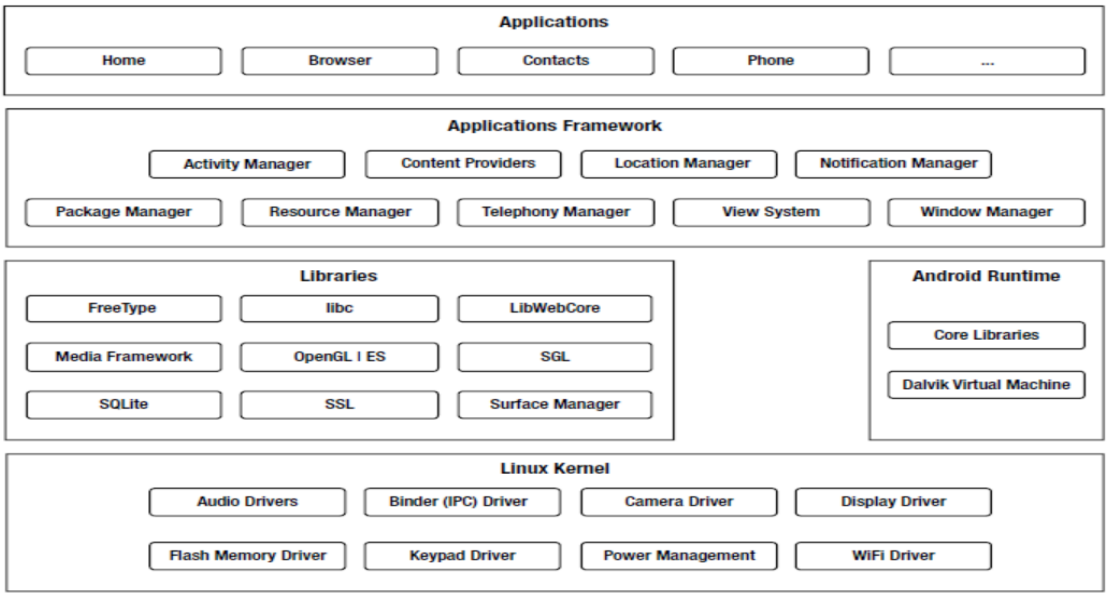

Figure (2.4): *This layered architecture.* [15]

1. Linux Kernel.

2. Android Libraries.

3. Android Runtime.

4. Android Framework.

5. Android Applications.

2.9 Previous study:

N	Paper Name	Date	Publisher/ author	Techniques
1	[3] SMS Encryption using AES Algorithm on Android	2012	International Journal of Computer Applications No.19. July 2012/ Rohan Rayarikar/ B.E in Computer Engineering	developed an application on Android platform
2	[4] Secured Mobile Messaging for Android application by using 3D-AES, PGP and Stegnography	2015	International Journal of Innovative Research in Computer and Communication Engineering/ Namrata A. Kale, Prof. S. B. Natikar, Priyanka D. Navgire	3D-AES which generates a symmetric key by shuffling the original key array three times and making the key better each time it is shuffled. - PGP for encryption and Compress the encrypted message to reduce its length, using Shannon fano algorithm technique.
3	[5] DATA ENCRYPTION USING BIO MOLECULAR INFORMATION	2014	International Journal on Cryptography and Information Security (IJCIS), Vol. 4, No. 3, September 2014 // BehnamBazli, Mustafa Anil Tuncel and David Llewellyn-Jones	In this work its take inspiration from DNA encryption schemes and use of biological alphabets to manipulate information by employing the DNA sequence reaction, to autonomously make a copy of its threads as an extended encryption key. And use of chemical properties of the DNA sequences of the cipher text to encrypt data over the public channel to add key extension and complexity to the encryption algorithm.
4	[6] Data Security in M-Learning Messaging Services	2011	INTERNATIONAL JOURNAL OF COMPUTERS AND COMMUNICATIONS/ Cătălin Boja, Paul Pocatilu AlinZamfiroiu	They implementations m-learning applications/ using Java ME and .NET CF platforms. To encryption for sensitive data (e.g. personal information, passwords, and marks)
5	[7] Building web applications on top of encrypted data using Mylar	2016	MIT CSAIL and †Meteor Development Group/ Raluca Ada Popa, Emily Stark, Jonas Helfer, Steven Valdez, Nickolai Zeldovich, M. Frans Kaashoek, and Hari Balakrishnan	presents Mylar, a platform that provides end-to-end encryption to web applications.

6	[8] New Approaches to Encrypt and Decrypt Data in Image using Cryptography and Steganography Algorithm	2016	International Journal of Computer Applications (0975 – 8887) Volume 143 – No.4, June 2016/Ako Muhammad Abdullah,Roza Hikmat Hama Aziz MSc. Computer Science University of Sulaimani Kurdistan Region-Iraq,	Built a new system to embed a secret message in a cover image using Hash based (3, 3, 2) LSB insertion method with Affine cipher algorithm
7	[9] Design of a secure android chatting application using end to end encryption	2017	JOURNAL OF SOFTWARE ENGINEERING & INTELLIGENT SYSTEMS ISSN 2518-8739 30th April 2017/ 1Ammar H. Ali, 2Ali MakkiSagheer	a secure chatting application with end-to-end encryption for smart phones, that used the android OS.a system is developed that provides end -to-end chatty application running on mobile devices that operate on android platform.

N	Results	Open issues
1	This application guarantees secure end to end transfer of data without any corrupt data segments.	The competitive edge and other benefits of mobility can be lost if smart phones and tablet PCs are not adequately protected against mobile device security threats.
2	Developed technique combines. It encrypts the SMS using 3D-AES algorithm and PGP. Since encrypted SMS compressed using a lossless algorithm, Shannon Fano algorithm. This approach is giving security results more than 95% in terms of precision and 97.43% in terms of recall.	In an asymmetric key system, Bob/sender and Alice/receiver have separate padlocks, instead of the single padlock with multiple keys from the symmetric example. This key send and received make padlock and which is much more complicated Alice asks Bob to send his open padlock to her through regular mail, keeping his key to himself.
3	This technique will add security to the key and message exploitation. Furthermore, if the cipher text is accessed and content is revealed, the true meaning of the message will not be revealed without the key.	This technique will enhance the security of the encryption mechanism by substitution, manipulation, and complexity. Furthermore, this technique can be used in many applications of information and

	This technique will enhance the security of the encryptions and algorithms	communication systems as well as adding more complexity to existing Encryption algorithms.
4	The study has been conducted on using this technology to increase the quality and the output of the educational process. A distinct approach is to use mobile devices to deliver educational content in anytime	If someone who needs to change the source code does not understand it due to lack of readability, they are forced to rewrite it and thus losing valuable time
5	Mylar protects the confidentiality of sensitive data fields against attackers that gained access to servers. Mylar stores sensitive data encrypted on the server, and decrypts that data only in users' browsers.	this approach suffers from three significant security, functionality, and efficiency short comings. First, a compromised server could provide malicious client-side code to the browser and extract the user's key and data, because a web application consists of many files, such as HTML pages, JavaScript Second, this approach does not provide data sharing between users, because a compromised server can supply arbitrary keys to users, and thus trick a user into using incorrect keys
6	This system ensures the data is secured against intruders within the network environment, by using Cryptography is a method to encrypt data and steganography is the art and science of hiding secret message in a cover image	If It have big data encrypted and It want to hidden in the image this date effect to the quality of image
7	End to End Encryption achieved by involving ECDH key exchange to provide the key pair (private and public) which will be exchange between the two parties to generate the secure shared key that will be used as a key for the encryption algorithms	

2.10 User interface design principles:

This part of user interface design principle and mobile interface design principles, based on mobile consultant Jonathan Stark. Before presents principles of mobile interface design and go any further, let's define the most common user interface design principles: [10]

- **The structure principle:**

Design should organize the user interface purposefully, in meaningful and useful ways based on clear, consistent models that are apparent and recognizable to users, putting related things together and separating unrelated things, differentiating dissimilar things and making similar things resemble one another. The structure principle is concerned with overall user interface architecture.

- **The simplicity principle:**

The design should make simple, common tasks easy, communicating clearly and simply in the user's own language, and providing good shortcuts that are meaningfully related to longer procedures.

- **The visibility principle:**

The design should make all needed options and materials for a given task visible without distracting the user with extraneous or redundant information. Good designs do not overwhelm users with alternatives or confuse them with unneeded information.

- **The feedback principle:**

The design should keep users informed of actions or interpretations, changes of state or condition, and errors or exceptions that are relevant and of interest to the user through clear, concise, and unambiguous language familiar to users.

- **The tolerance principle:**

The design should be flexible and tolerant, reducing the cost of mistakes and misuse by allowing undoing and redoing, while also preventing errors wherever possible by tolerating varied inputs and sequences and by interpreting all reasonable actions.

- **The reuse principle:**

The design should reuse internal and external components and behavior of users, maintaining consistency with purpose rather than merely arbitrary consistency, thus reducing the need for users to rethink and remember

2.11 Mobile interface design principles:

Mobile Consultant Jonathan Stark outlined 10 design principles for the mobile interface based on his work shops, Jonathan compiled the top principles of mobile interface design in the following: [10]

1. Mobile mindset:

Because of the differences between mobile and desktop, it's imperative to get yourself into a mobile mindset before getting started. With the following characteristics:

- Be focused.
- Be unique.
- Be charming.
- Be considerate all different needs.

2. Mobile contexts:

The image of the busy professional racing through the airport with a bag in one hand and smart phone in the other is what lots of people picture when they think about mobile computing context. It is certainly one context, but it's not the only one, users need to consider three major mobile contexts: (Bored, Busy and Lost)

3. Global guidelines:

Different applications call for different approaches, designs and techniques. That said, the inherent nature of a pocket-sized touch screen device suggests several global guidelines; i.e., the stuff that always matters:

I. Polish.
II. Targets.
III. Content.
IV. Thumbs.
V. Controls.
VI. Scrolling.
VII. Responsiveness.

4. Navigation models:

There are plenty of novel navigation models for mobile apps (Path's radial corner navigation springs to mind) but if going to use one of the common navigation models, be sure to pick the one that makes the most sense for app.

I. None.
II. Tab bar.
III. Drill down.

5. User input:

Typing stinks even on the best devices, so, should do what can to make it easier for your users.

6. Gestures:

One of the most iconic aspects of modern touch interfaces is that they support gesture-based user interaction. As cool as gestures are, there are several things you need to keep in mind:

I. Invisible.
II. Two hands.
III. Nice to have.
IV. No replacement.

7. Orientation:

Portrait is by far the most popular orientation, so optimise for this case first.

8. Communications:

Communication provide:

I. Feedback.
II. Modal alerts.
III. Confirmations.

9. Launching:

When a user goes back into your app after having used it previously, they should resume operations right where they left off. This will give the illusion of speed and contribute to an overall feel of responsiveness.

10. First impressions:

I. Your icon: Icon has to compete for attention in a sea of other icons.

II. First launch: First launch is a make or break situation.

If app provides complex functionality, might want to include a 'tips and tricks' overlay or perhaps a few panels of orientation screens.

Chapter Three

Methodology

CAPTER 3

METHODOLOGY ANALYSIS AND SOFTWARE REQUIREMENT

3.1 Introduction:

This chapter provides a detailed Methodology analysis and explanation of the implemented steps. This section begins with a review and presentation of the application flowchart, scenarios, the sequential diagram, application development and software and hardware requirements.

3.2 Flowchart:

The flowcharts section discusses the application flow diagram and scenario used to ensure secure text messages exchange using the mobile application. The scenario is illustrated as follows:

3.2.1 Transmitter end:

Create a new account for user, using the mobile application and obtain User name and Password. The application will do then allow:

1. Log in.
2. Generate random key, converting it to **QR** code and saved it.
3. Determine the destination of the message.
4. Write the text message to be sent.
5. The text message is sent after it has been encrypted using the **AES** algorithm.
6. Share the previously saved QR code, via an available application.

3.2.2 Receiver end:

In this part, the same steps (**1** and **2)** in the previous part are followed, and then the following is performed.

3. Receive the encrypted text message.
4. Scan the **QR** using one of the following tools.
    I. The CAMERA.
    II. The GALLERY.
5. Decrypt and open the text messages.

3.2.3 The device sending the text message:

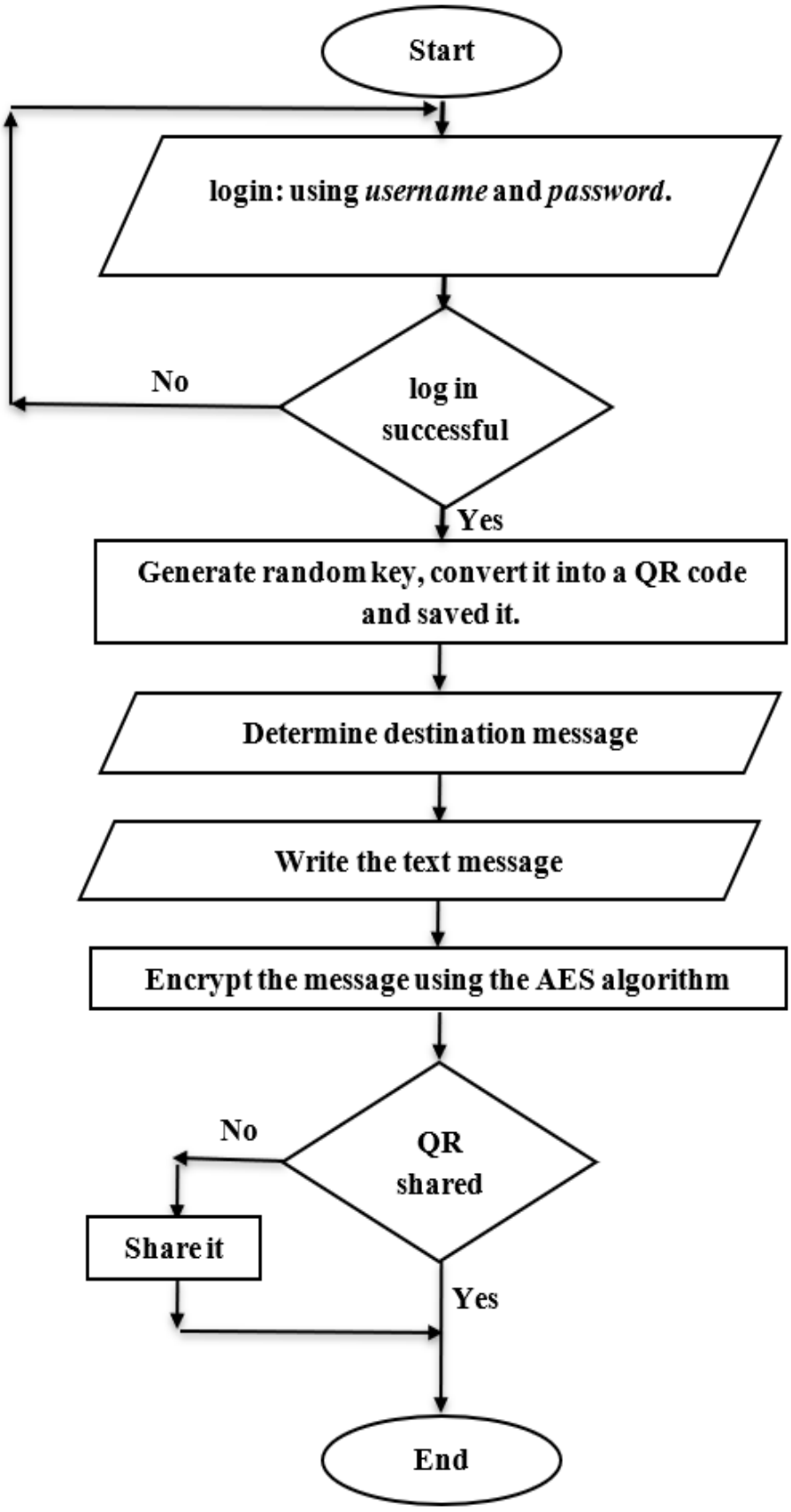

Figure (3.1): *Massage transmission flow chart.*

Step 1: Start:

The application starts with fulfilling the conditions below:

I. Full registration requirements, first, second name, then password and confirmation.
II. Fulfill the requirements of **Step 2**.

Step 2: Login:

Log in with your username and password for the mobile app. The user must previously be registered in the application's database, if the input does not match the information in the database, the user is denied access.

Step 3: Generate key:

The application generates a random key based on user name and password using **AES** algorithm. The key is saved after converting it to QR code format.

Step 4: The destinations:

Determine message destination, who is receiving the message.

Step 5: Write message:

Write or enter the message text to be sent to the recipient.

Step 6: Send message:

Encrypt the message using the AES algorithm and send it.

Step 7: Share QR code:

Confirm the QR code has been shared.

Step 8: End:

End steps of the sender.

3.2.4 The device receiving the text message:

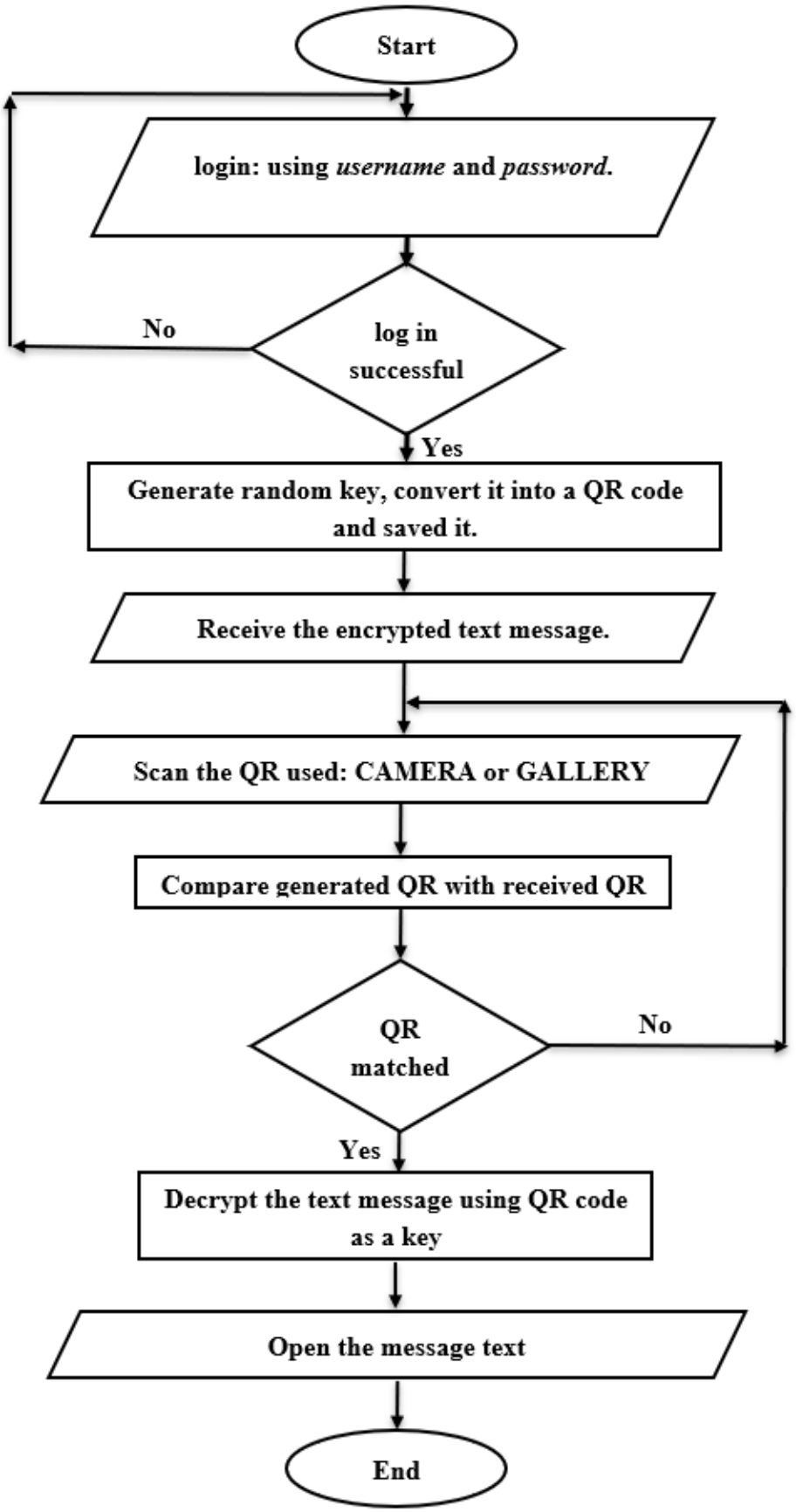

Figure (3.2): *Massage receiver flow chart.*

Step 1: Start:

The application starts.

Step 2: Login:

Provide login by entering user name and password. However, the app's features cannot be accessed without the correct login data, the following steps must be followed:

I. Receive the QR code.
II. Receive the text message.

Step 3: Generate key:

In receiver part also, random key is generated to be used later as a key to encrypt text messages.

Step 4: Receive message:

Receive the encrypted text message.

Step 5: Scan QR code:

Scan the QR received via other apps, using: CAMERA or GALLERY.

Step 6: QR matched:

If the QR code matches the code received by the sender of the text message, the code is converted into a key to be used to decrypt the message.

Step 7: decrypt message:

Decrypt the text message using QR code as a key and open message.

Step 8: End:

End steps of the receiver.

3.3 Sequence diagram:

Below is a comprehensive explanation of the sequence diagram for the implementation of the previous scenarios and integration of roles among all parties involved in completing the operation process (sender, receiver, provider and the system).

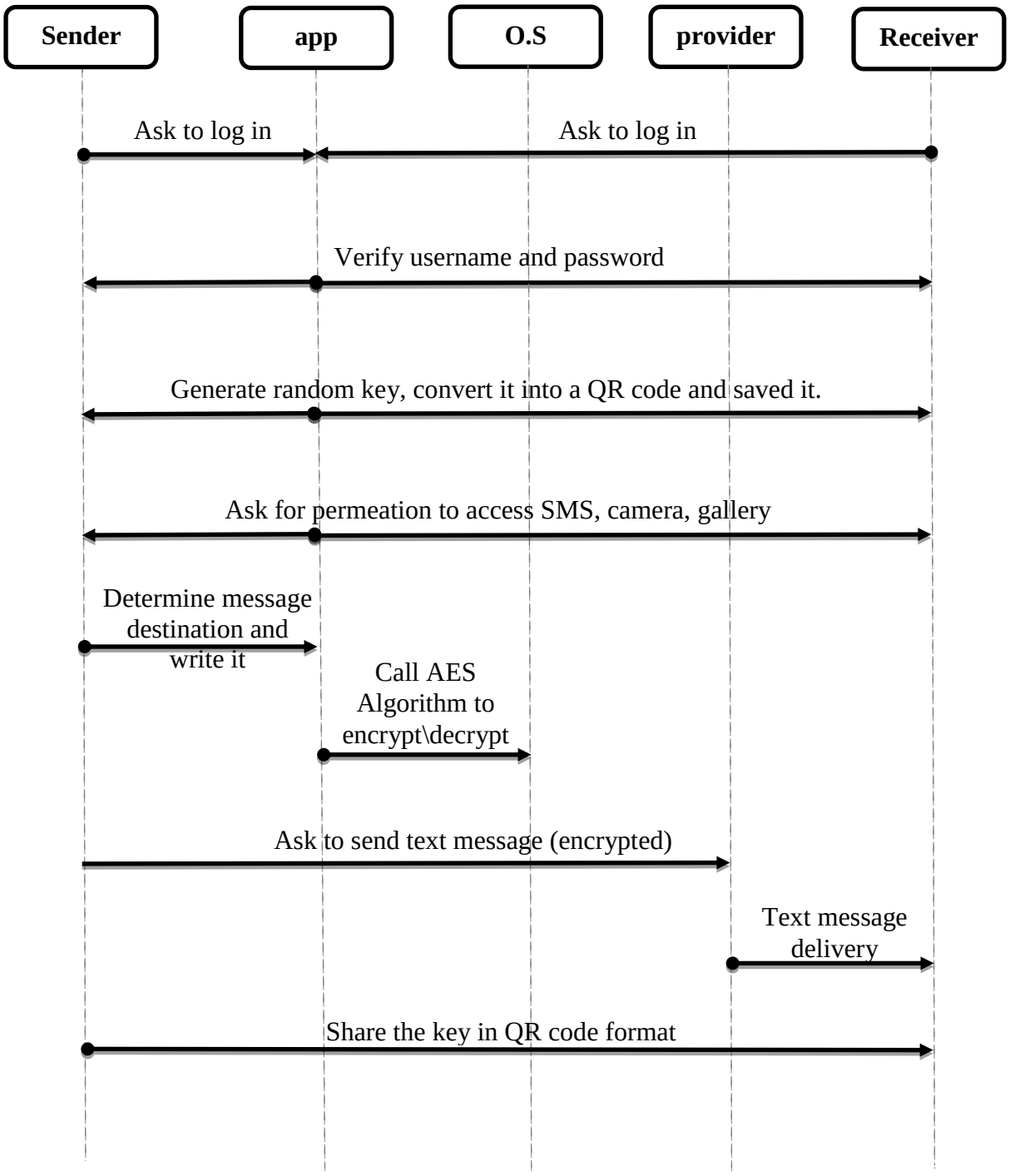

Figure (3.3): *The sequence diagram.*

3.4 Software requirement:

I. Android SDK:

Android software development SDK is the process by which applications are created for devices running the Android operating system. Google states that "Android apps can be written using Kotlin, Java, and C++ languages" using the Android software development kit SDK, while using other languages is also possible. All non-JVM languages, such as Go, JavaScript, C, C++ or assembly, need the help of JVM language code, that may be supplied by tools, likely with restricted API support. Some programming languages and tools allow cross-platform app support (i.e. for both Android and iOS). Third party tools, development environments, and language support have also continued to evolve and expand since the initial SDK was released in 2008. The official Android app distribution mechanism to end users is Google Play; it also allows staged gradual app release, as well as distribution of pre-release app versions to testers. [17]

II. Android ADT:

Android Development Tools ADT is a plugin for the Eclipse IDE that is designed to give you a powerful, integrated environment in which to build Android applications.

ADT extends the capabilities of Eclipse to let you quickly set up new Android projects, create an application UI, add components based on the Android Framework API, debug your applications using the Android SDK tools, and even export signed (or unsigned) APKs in order to distribute your application.

Developing in Eclipse with ADT is highly recommended and is the fastest way to get started. With the guided project setup it provides, as well as tools integration, custom XML editors, and debug output pane, ADT gives you an incredible boost in developing Android applications.

This document provides step-by-step instructions on how to download the ADT plugin and install it into your Eclipse development environment. Note that before you can install or use ADT, you must have compatible versions of both the Eclipse IDE and the Android SDK installed. For details, make sure to read Installing the ADT plugin, below.

If you are already using ADT, this document also provides instructions on how to update ADT to the latest version or how to uninstall it, if necessary. [18]

III. Android Studio:

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems or as a subscription-based service in 2020. It is a replacement for the Eclipse Android Development Tools (E-ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0.

On May 7, 2019, Kotlin replaced Java as Google's preferred language for Android app development. [13] Java is still supported, as is C++. [19]

3.5 Hardware requirement:

The material needs are summarized below:

1) Computer with a minimum of following requirements:
2) Processor: core i5, Hard Disk: 500 GB, Memory: 4 GB RAM
3) Two Android devices with minimum version 5.0 (Lollipop) API level 13.

3.6 User requirements:

1) Internet services.
2) SMS provides.

Chapter Four

Implementation and Results

CAPTER 4

DESIGN AND IMPLEMENTATION AND RESULTS

4.1 Introduction:

This chapter presents the design of the mobile application according to the methodology described in the previous chapter, presents the application user interface designs, and the results of the application are presented. It displays test results after running the application in practice.

4.2 User interface for mobile application:

The figure (4.1): Shows the shortcut icon for application after installing, it appears in the distinctive ring frame within the set of applications installed on the mobile device.

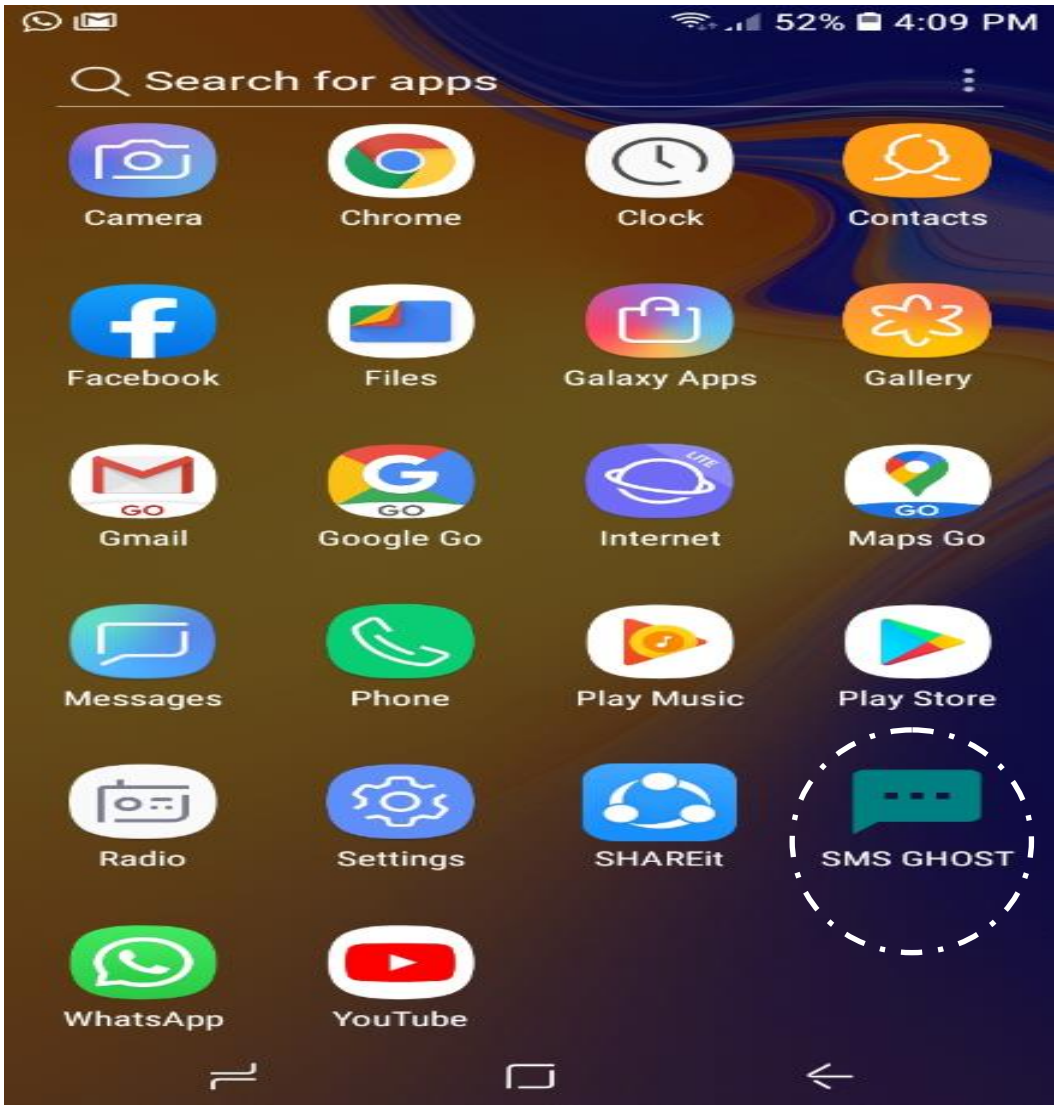

Figure (4.1): *The app icon.*

The figure (4.2): Shows registration screen for the application, this screen shows up when sign up first time.

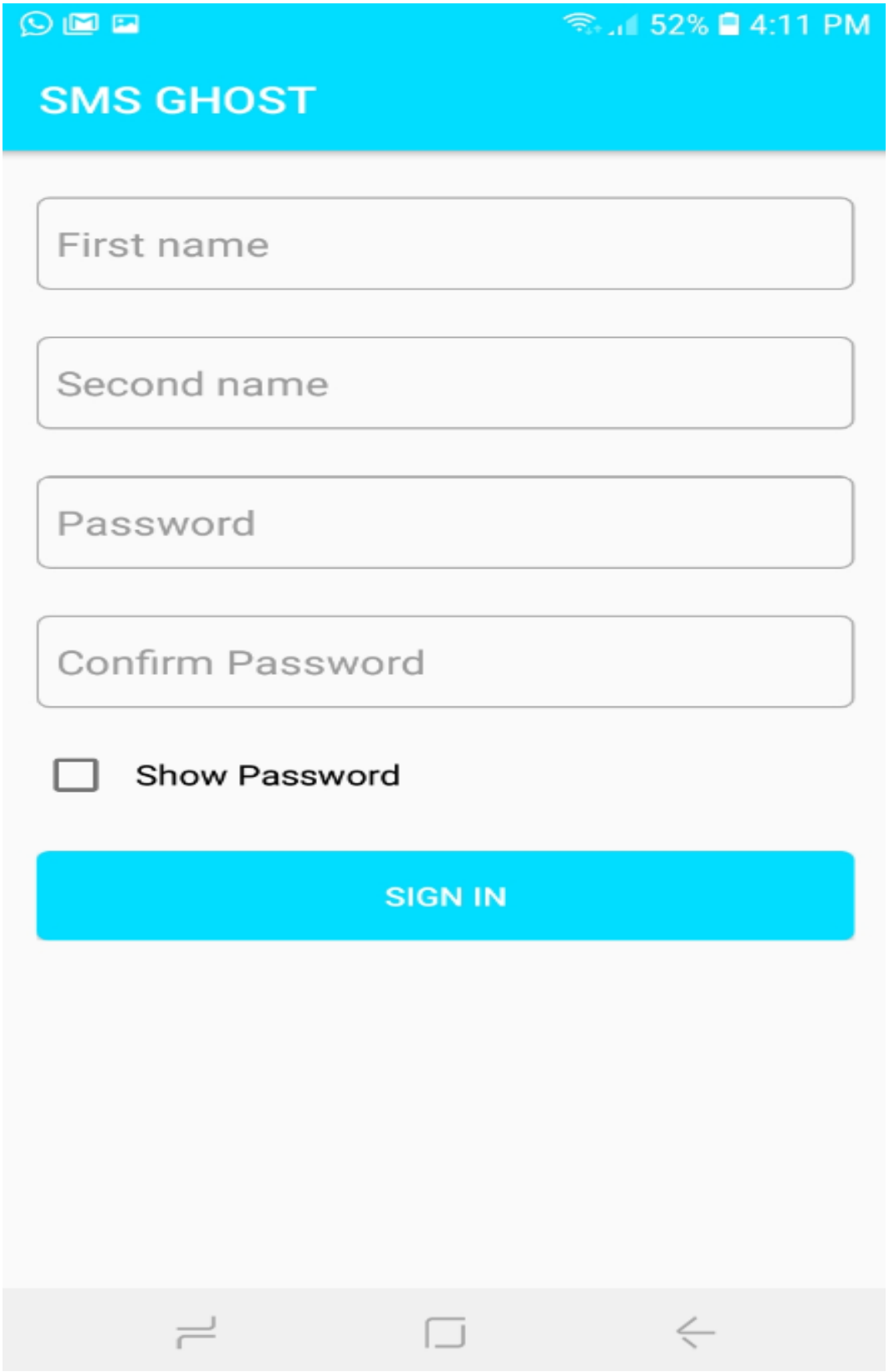

Figure (4.2): *registration screen.*

The figure (4.3): Shows the application's login screen, where the user's name and password are entered correctly and get to the next screen, unless either or both are incorrect. They must be corrected or no access to the application is granted.

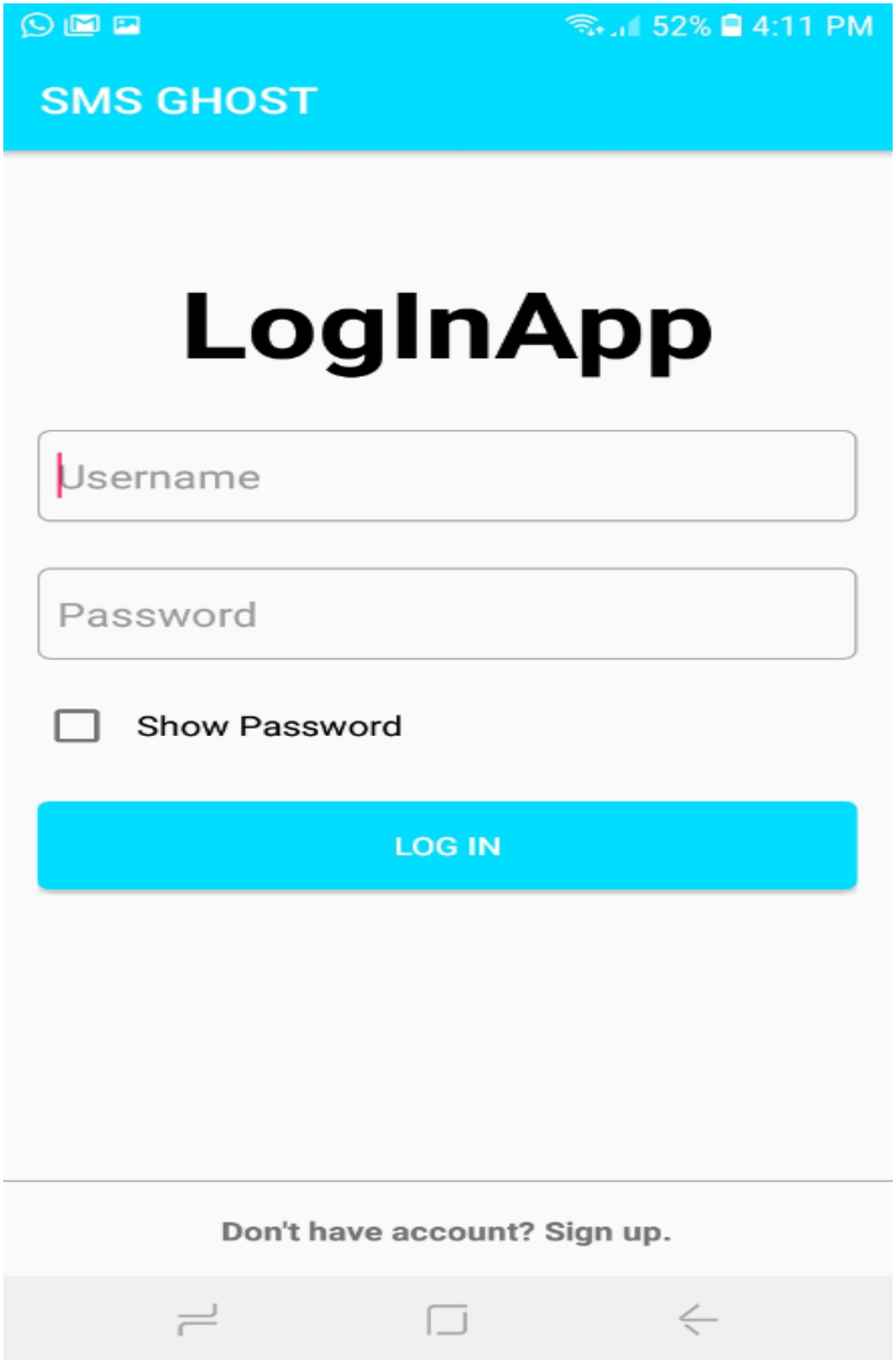

Figure (4.3): *Login screen.*

The figure (4.4): Shows the application's main screen, which displays the number of the contact intended to receive the text message, message content, share key button and send button it however, any sender using the application cannot successfully send a message unless his QR code is shared with the message receiver.

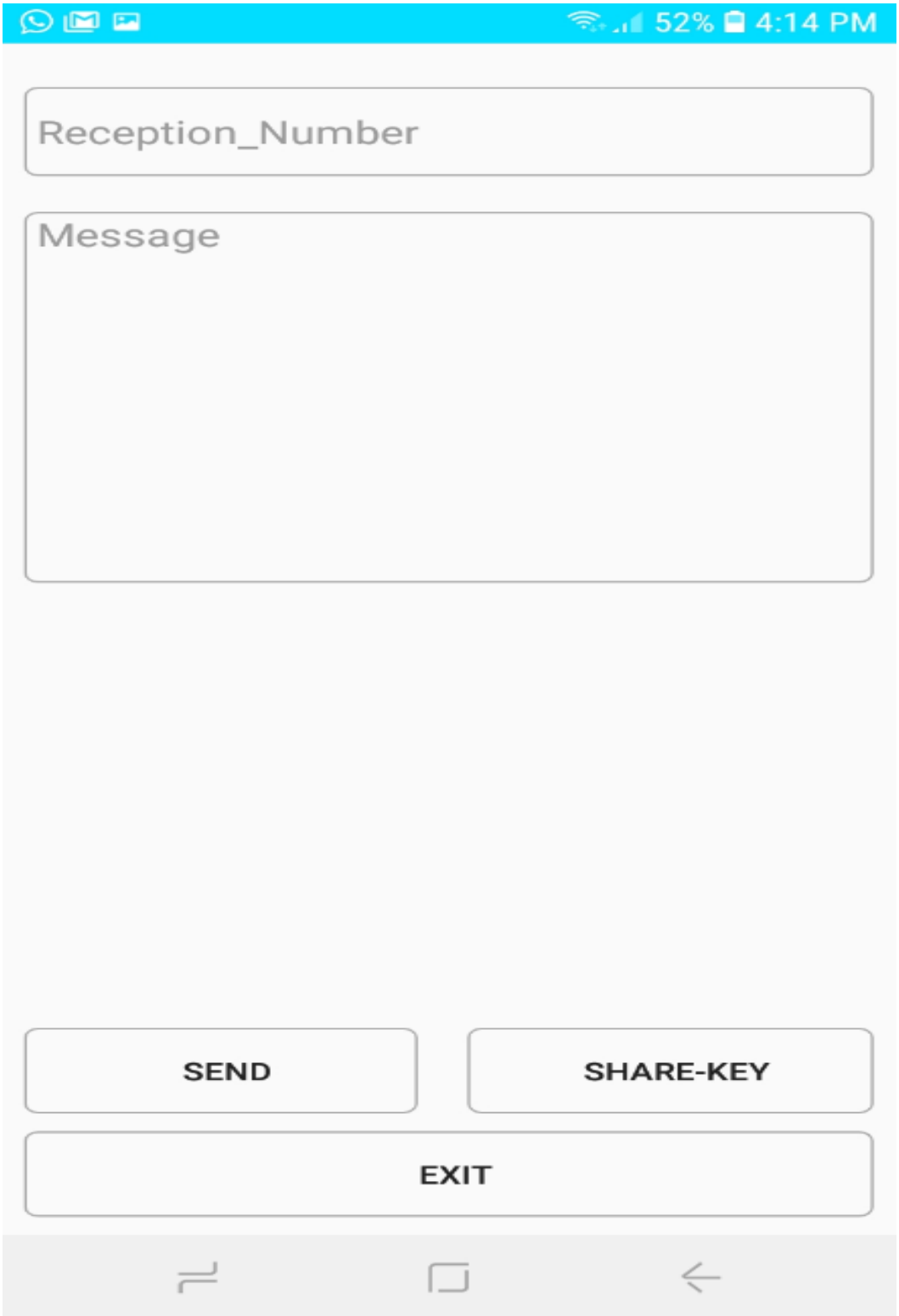

Figure (4.4): *The main screen.*

The figure (4.5): Shows the random key generation button, this screen is launched after a success log in.

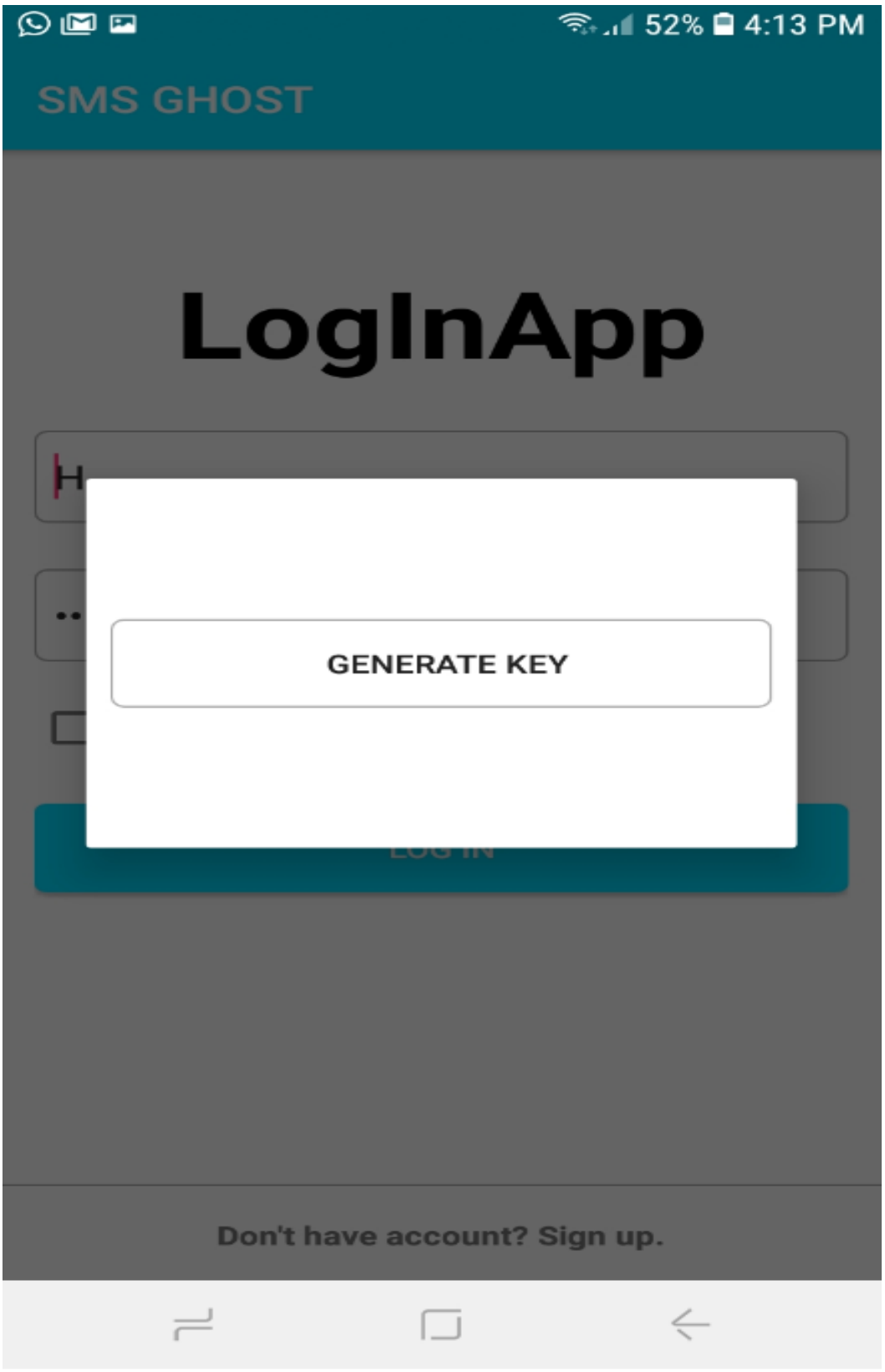

Figure (4.5): *The generate key screen.*

The figure (4.6): Shows the QR code which will be shared.

Figure (4.6): *share QR code.*

4.3 User interface for registration and setting:

The figure (4.7): Shows the application request to use the camera.

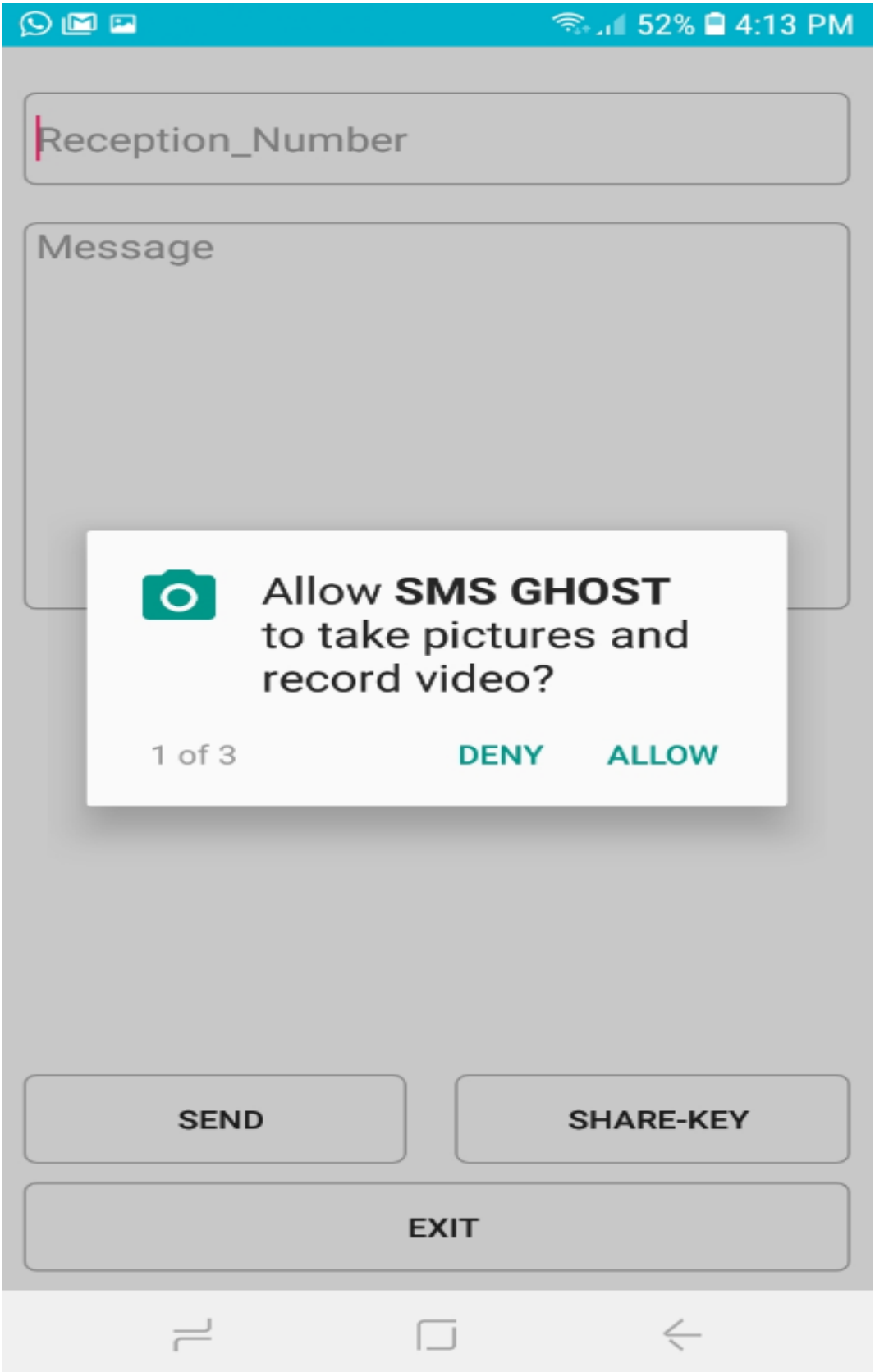

Figure *(4.7):* *application use camera.*

The figure (4.8): Shows the confirmation of user permission to use SMS.

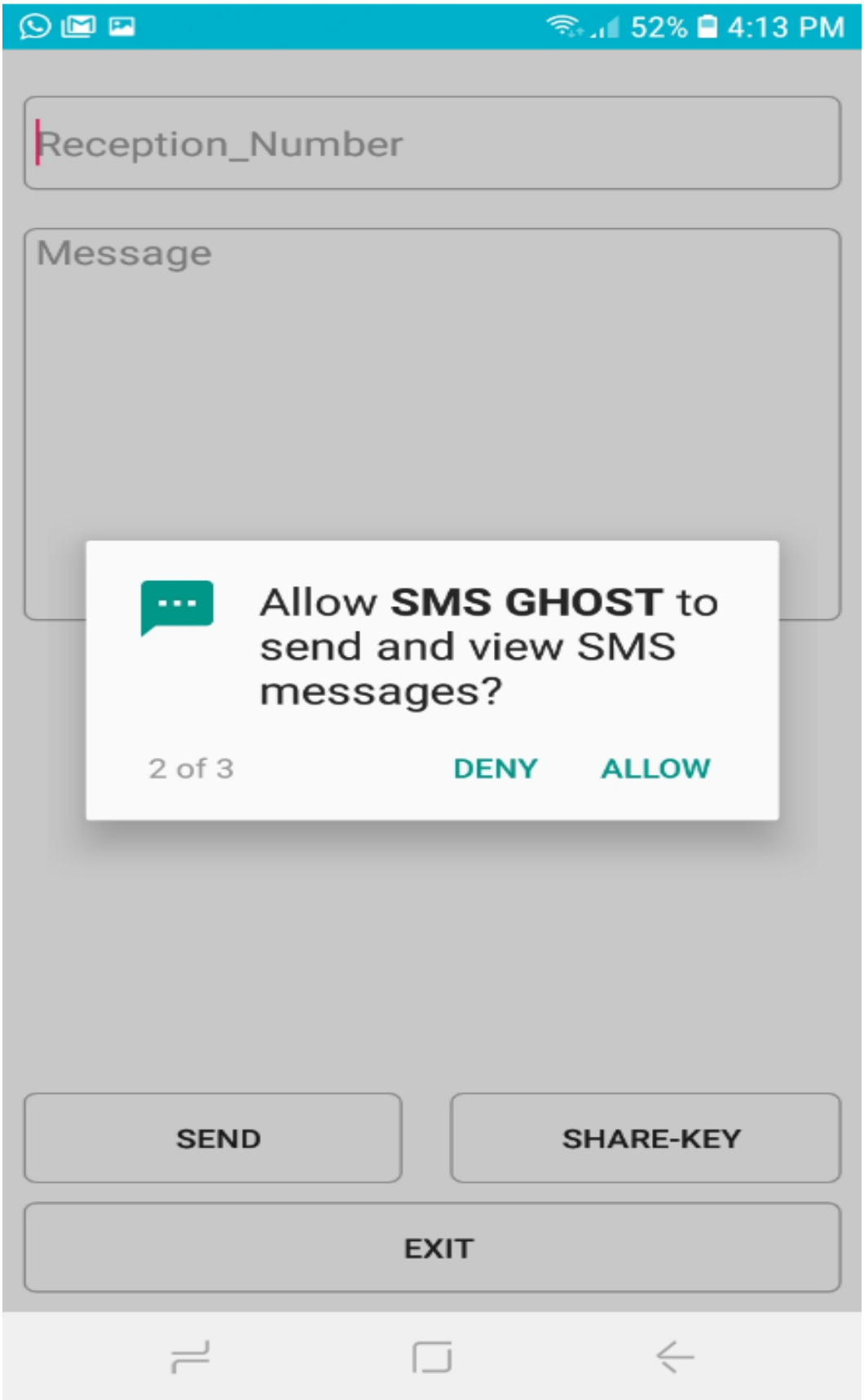

Figure (4.8): The access permission to use SMS

The figure (4.9): Shows the application request to access the gallery.

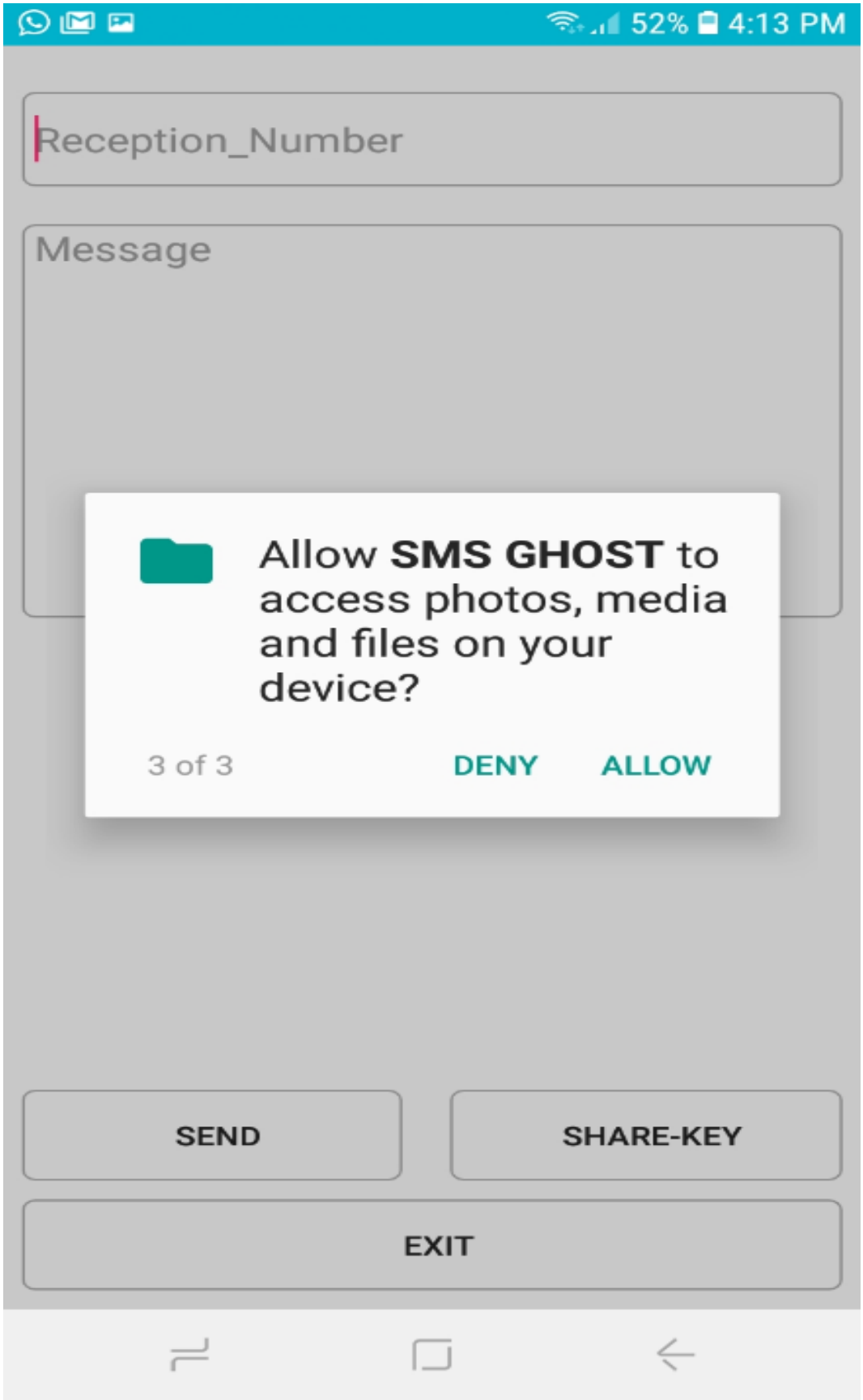

Figure (4.9): allow to access the gallery

4.4 Results of sender:

After completing the design process according to the approach mentioned in the previous chapter. And installing the application on the mobile phone, the following results appear:

Figure (4.10): shows the Application Installed Successfully screen.

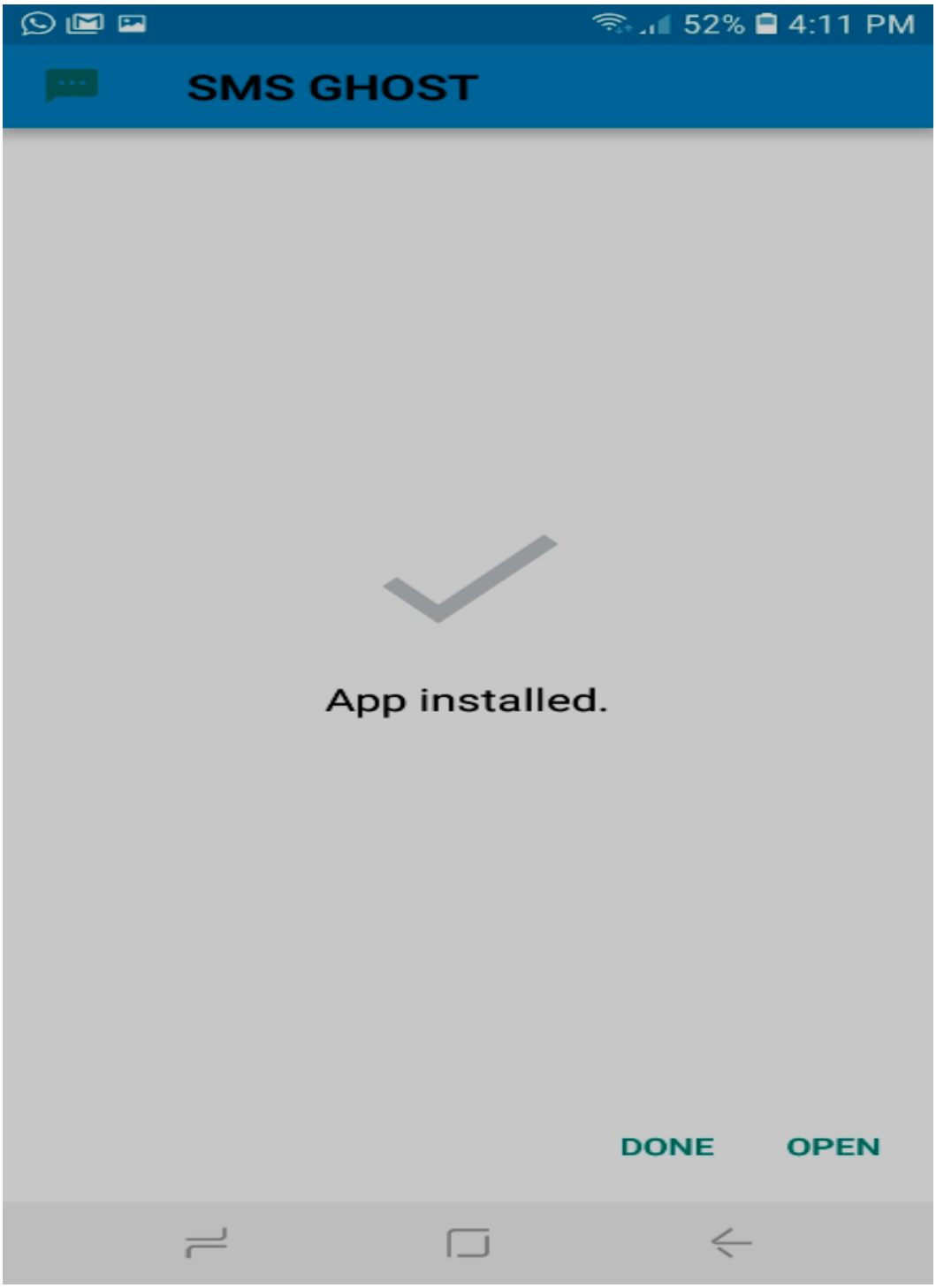

Figure (4.10): *The application installed successfully.*

When the application is launched, the login screen of the application is directly displayed. This requires the correct username and password to be entered without the slightest ambiguity for the previously registered user. As for the new user he must register first.

- If the application is installed successfully, it appears in the list of application manager as figure (4.11).
- If the registration is successful, the message shows as in figure (4.13)
- In the event that the user name or password is not entered correctly, the result will appear as shown in Figure (4.14).
- If the entries are correct, then the login will be done as in figure (4.15).

Figure (4.11): shows the application under the "Application Manager List". The application can be controlled by uninstall, in addition to knowing the size and features of the application.

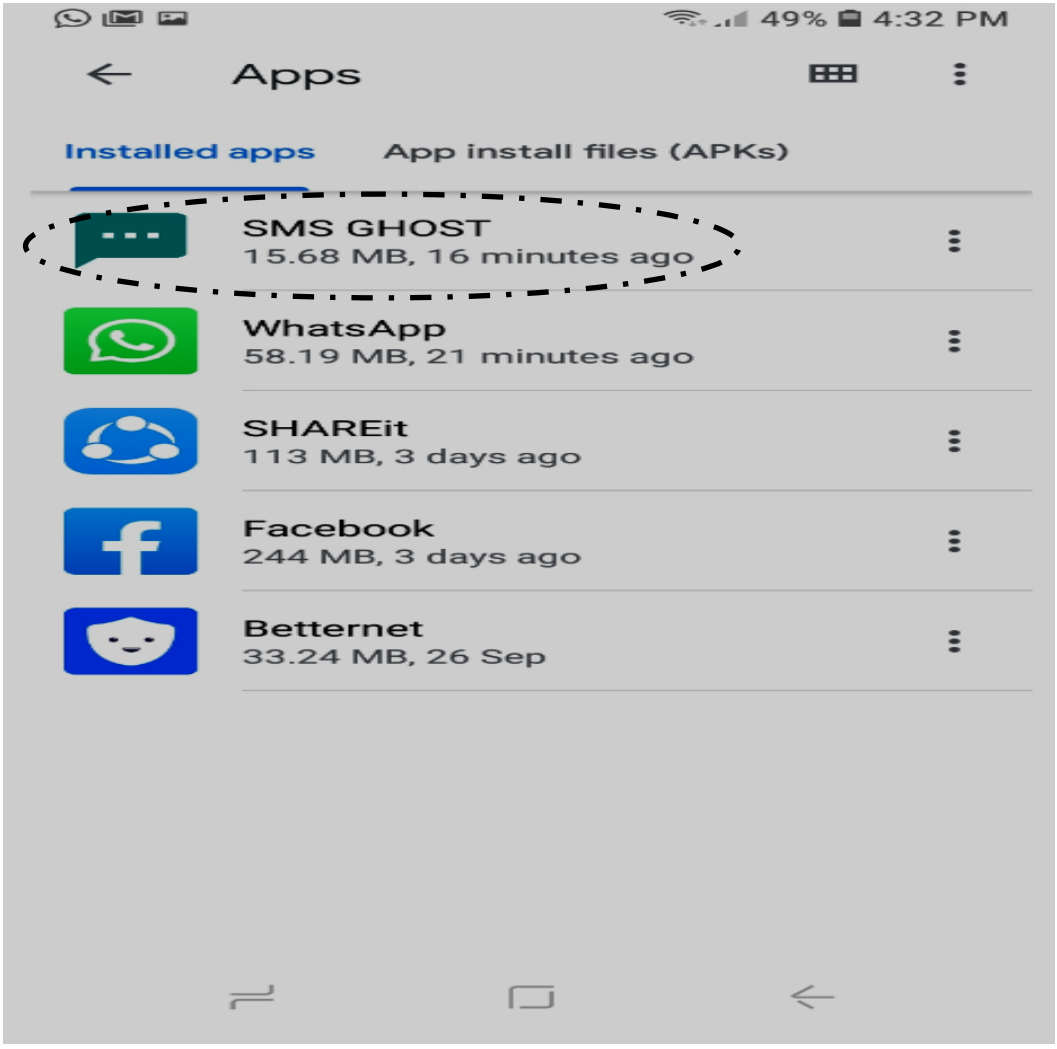

Figure (4.11): *The app in the Application list menu.*

Figure (4.12): The main registration screen shows the first, second name, password and confirmation.

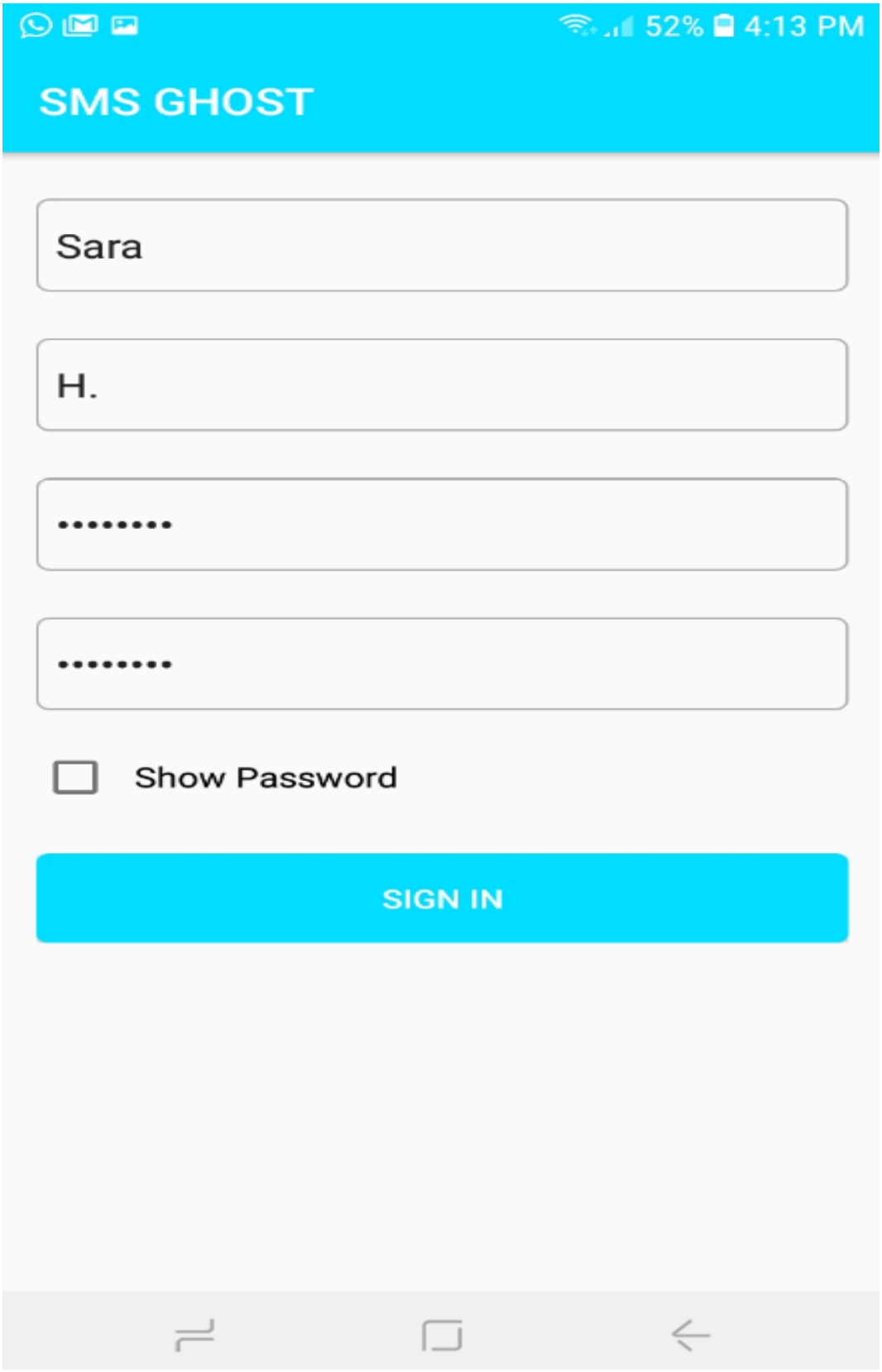

Figure (4.12): *The registration information screen.*

Figure (4.13): shows the login screen for the application after completing the registration information correctly, a confirmation registration message will appear here.

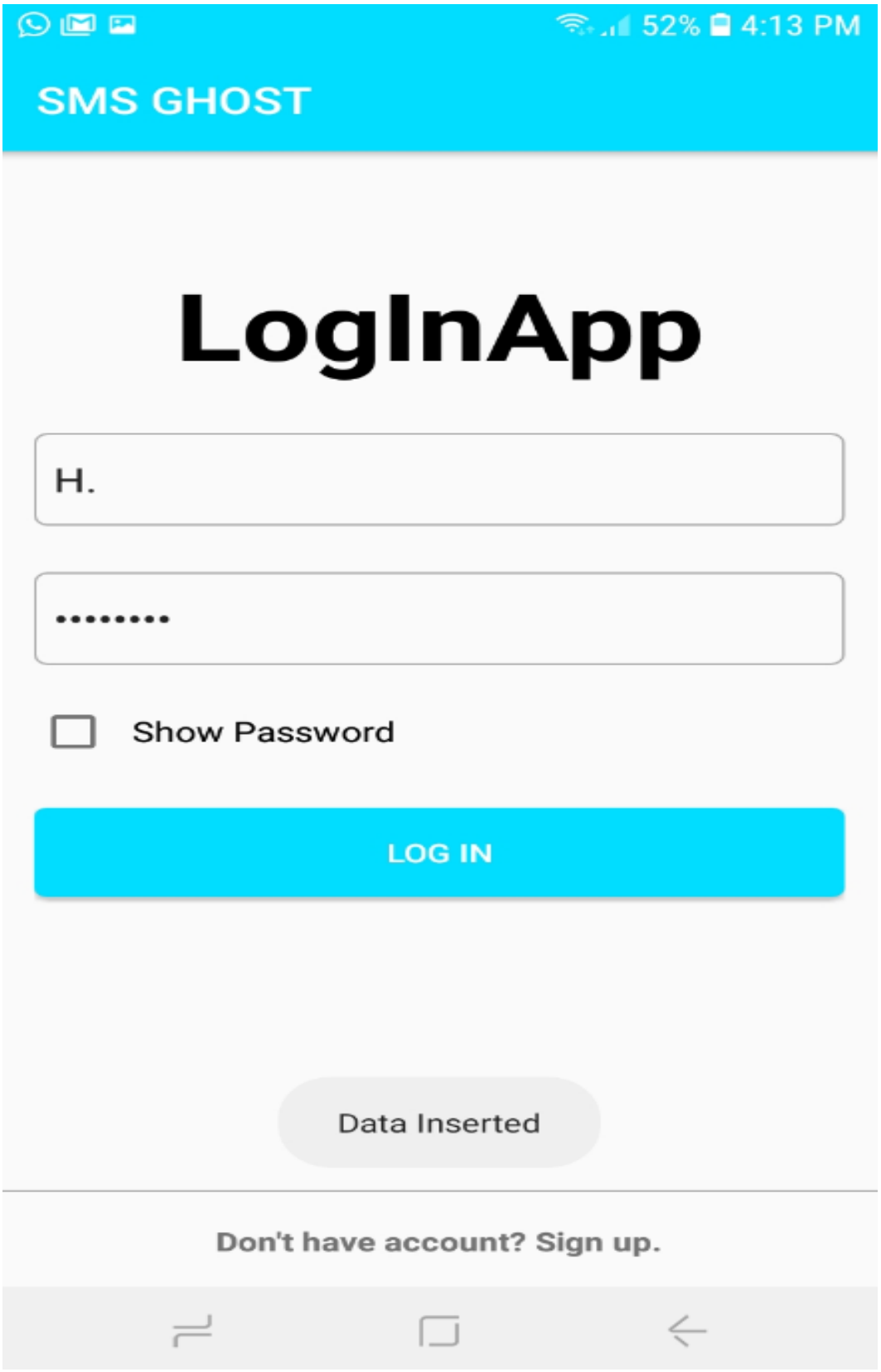

Figure (4.13): *The confirmation registration screen.*

Figure (4.14): shows the application's login screen if the username, password of either or both of them are entered incorrectly, in this case a message is displayed stating that there is an error.

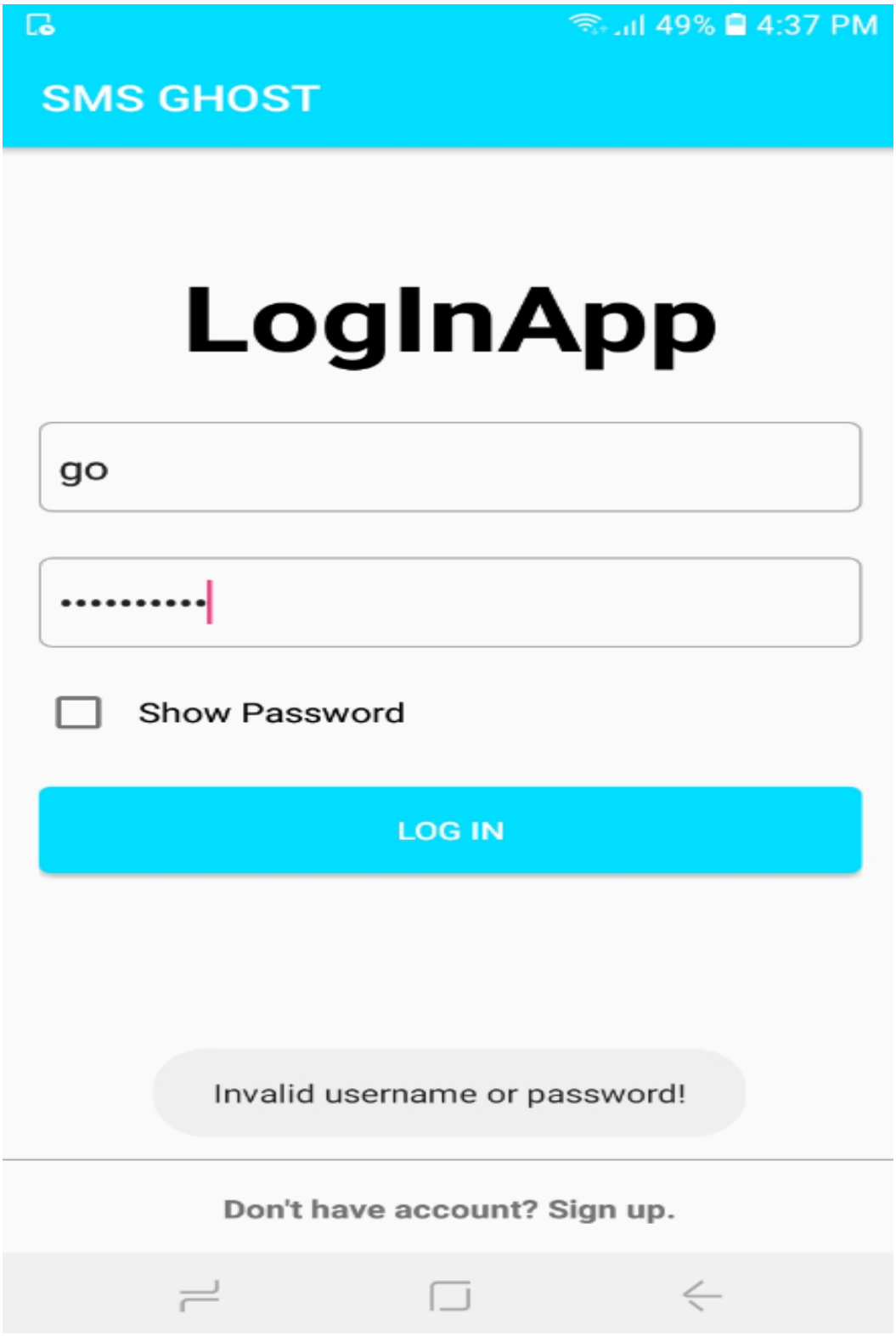

Figure (4.14): *The login incorrect screen.*

The figure (4.15): Shows the application login screen, if any user's name, password, or both entered incorrectly, a warning message will be generated indicating an error.

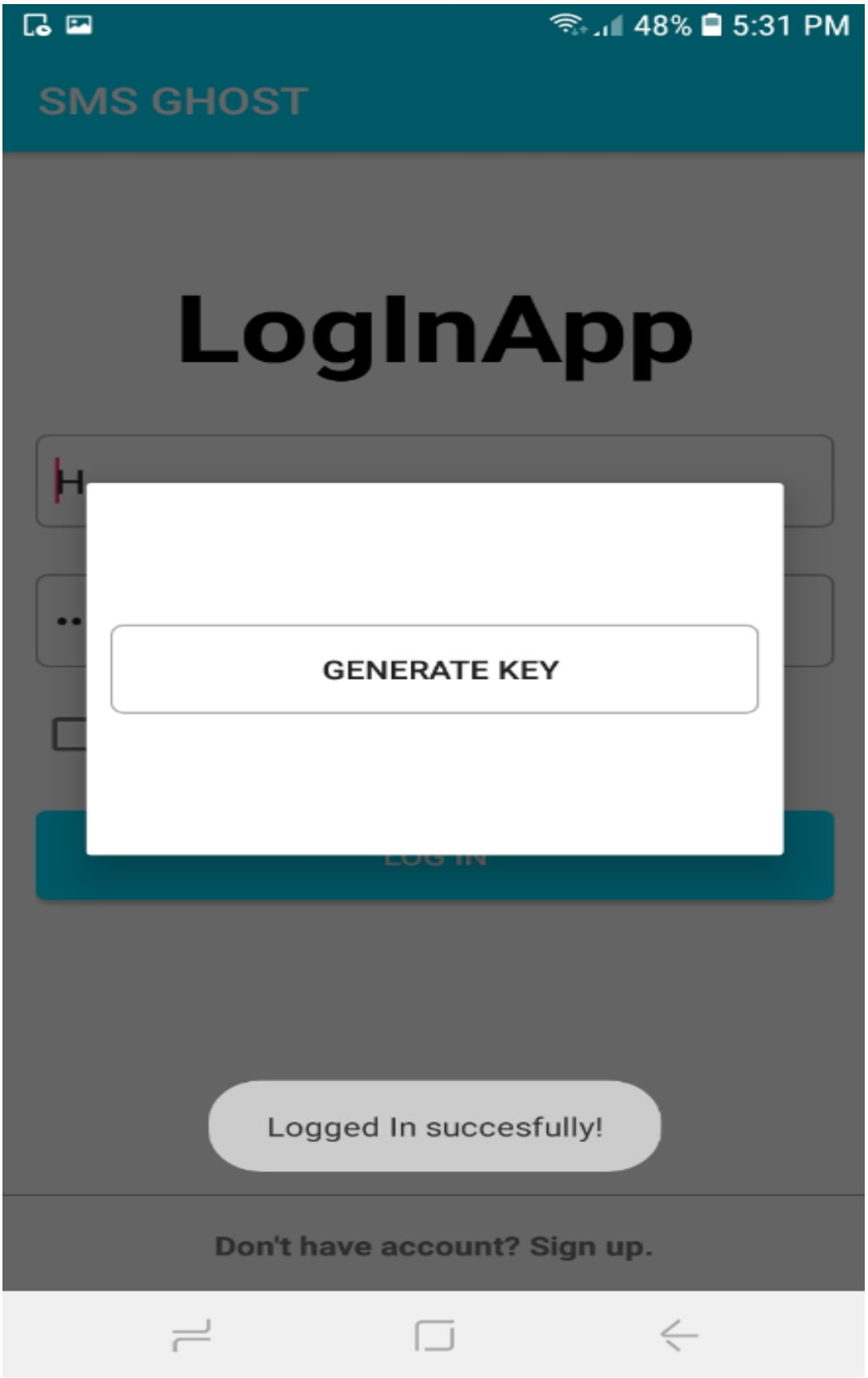

Figure (4.15): *The login successful screen.*

Figure (4.16): shows the completion and success screen of granting permissions to the application to use SMS, camera, in addition to accessing the gallery.

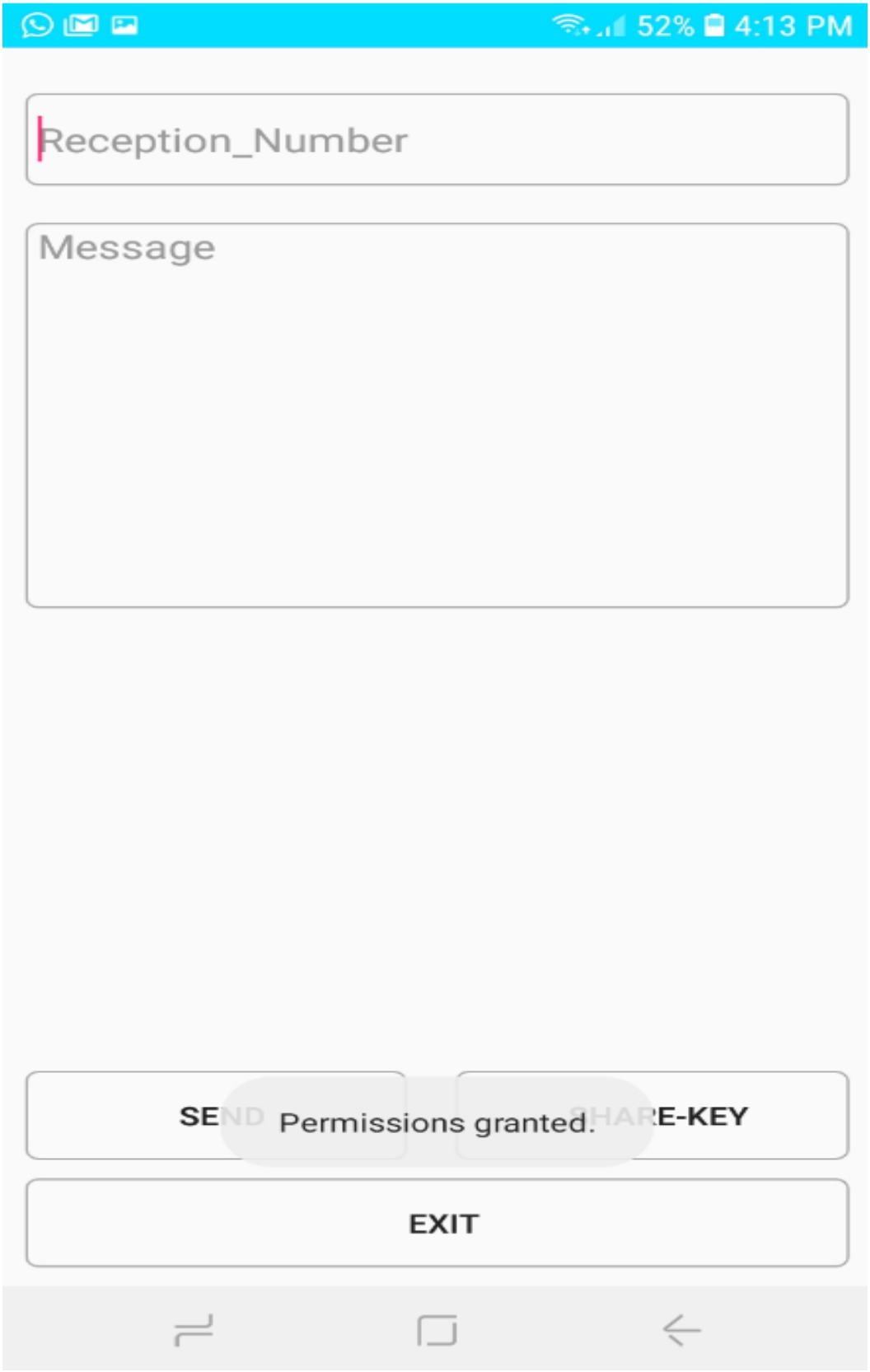

Figure (4.16): *The permission granted.*

Figure (4.17): shows the success and completion screen of sending the text message after determining the destination of the message and its content.

Figure (4.17): *The message sent successful.*

Figure (4.18): shows the key sharing screen using any of the popular social media applications as another party to increase security.

Figure (4.18): *Share the QR as a key.*

4.5 Results of receiver:

After sending the text message from the source and sharing the key with the message recipient, the following occurs:

- Download and save the QR code from social media applications.
- Scan the QR code and use it to open the text message.

The figure (4.19): Shows the text and key arriving at the recipient.

Figure (4.19): *The text and key arrive.*

The figure (4.20): Shows the QR in the gallery.

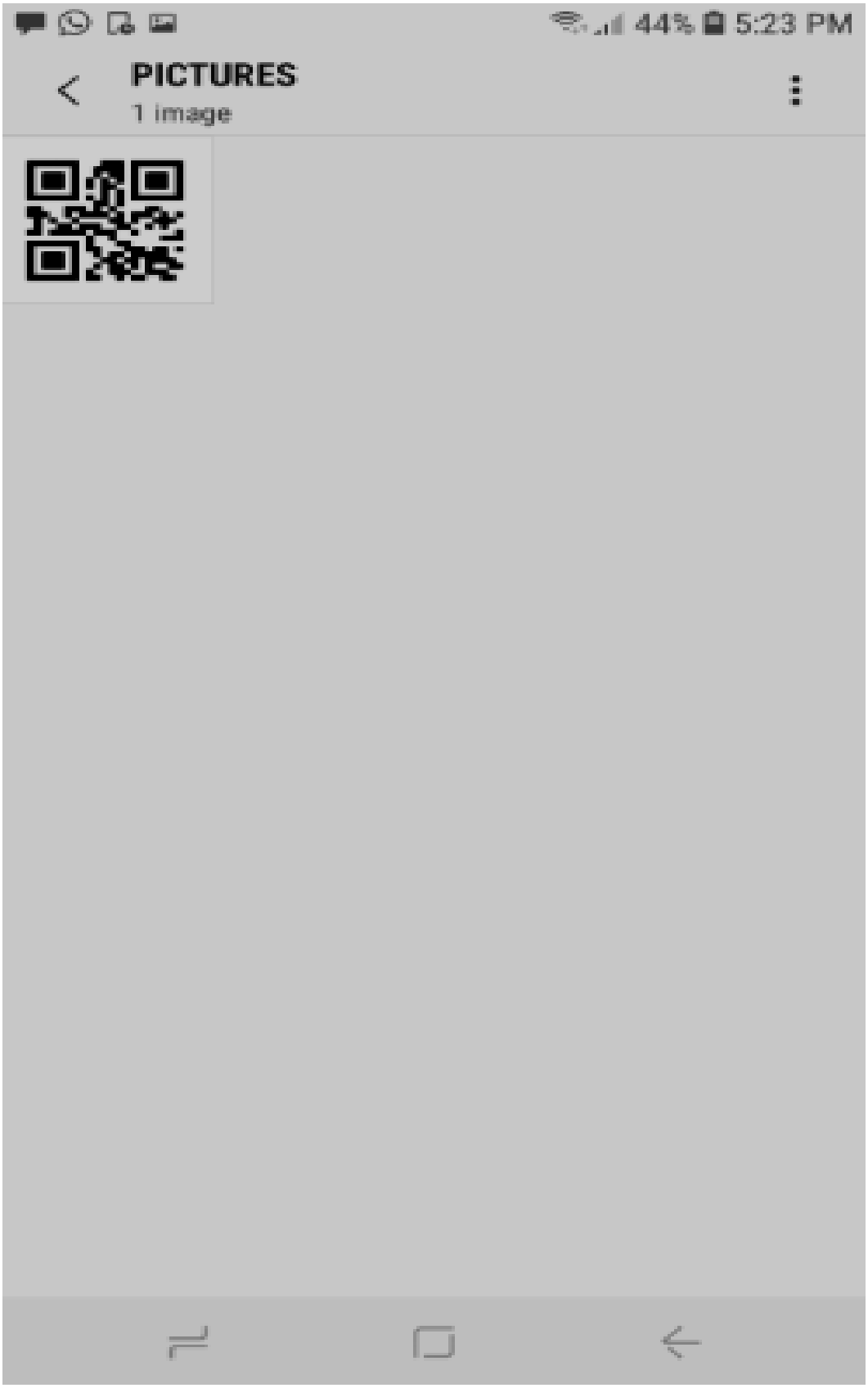

Figure (4.20): *The QR in the gallery.*

The figure (4.21): Shows the scan of the QR code screen.

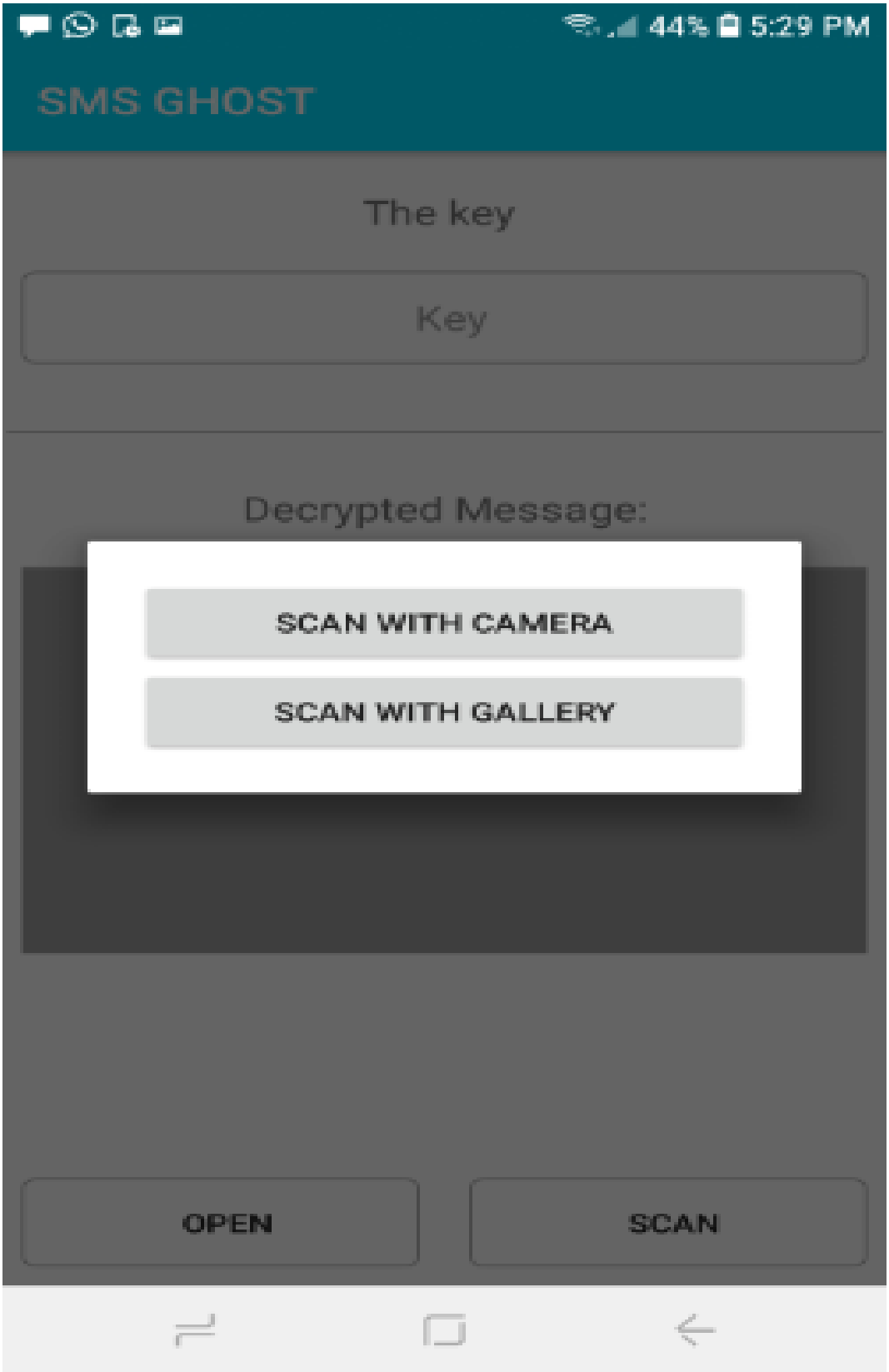

Figure (4.21): *The QR scan.*

The figure (4.22): Shows the message body before opening it.

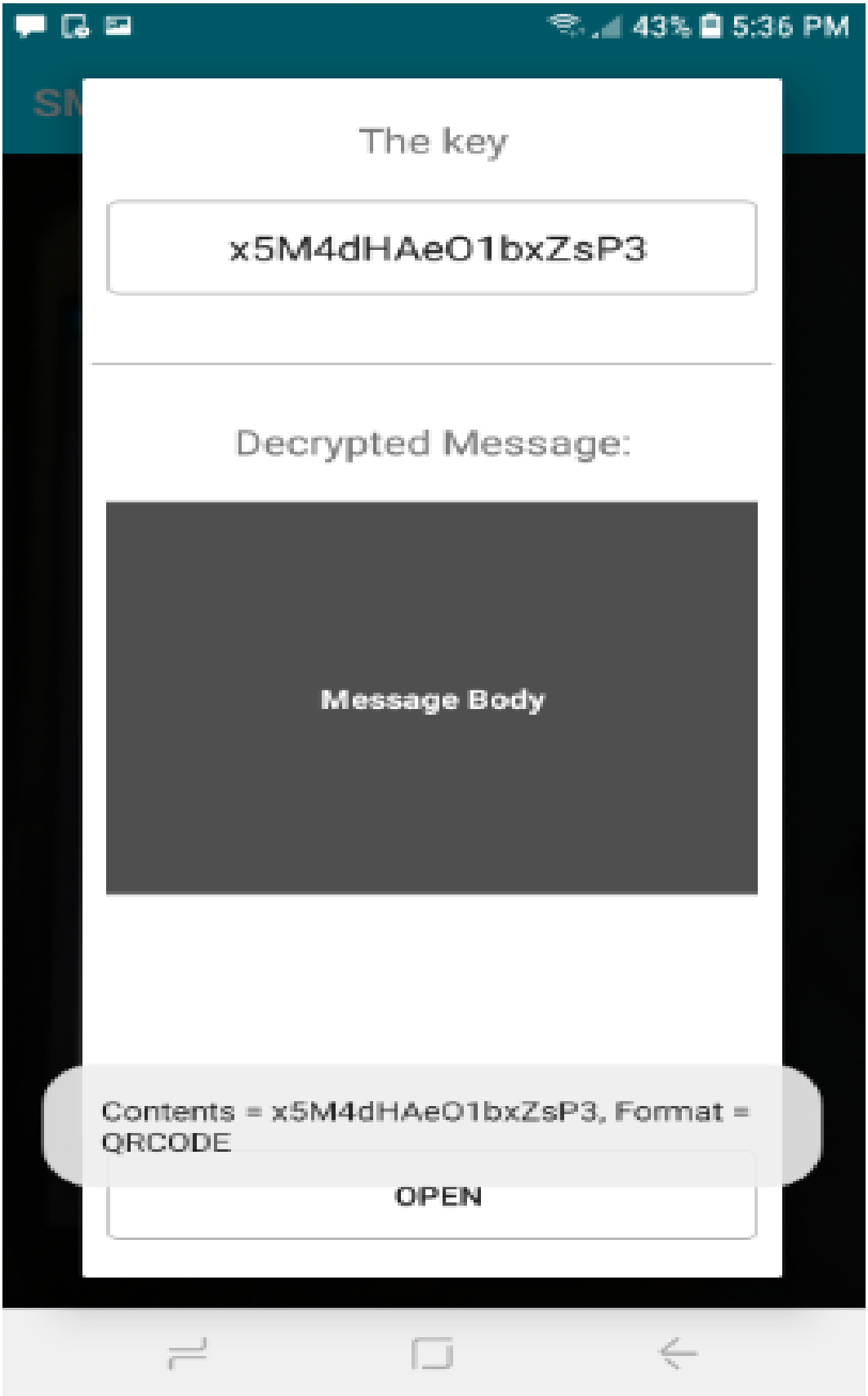

Figure (4.22): *The message body.*

The figure (4.23): Shows the original message after process.

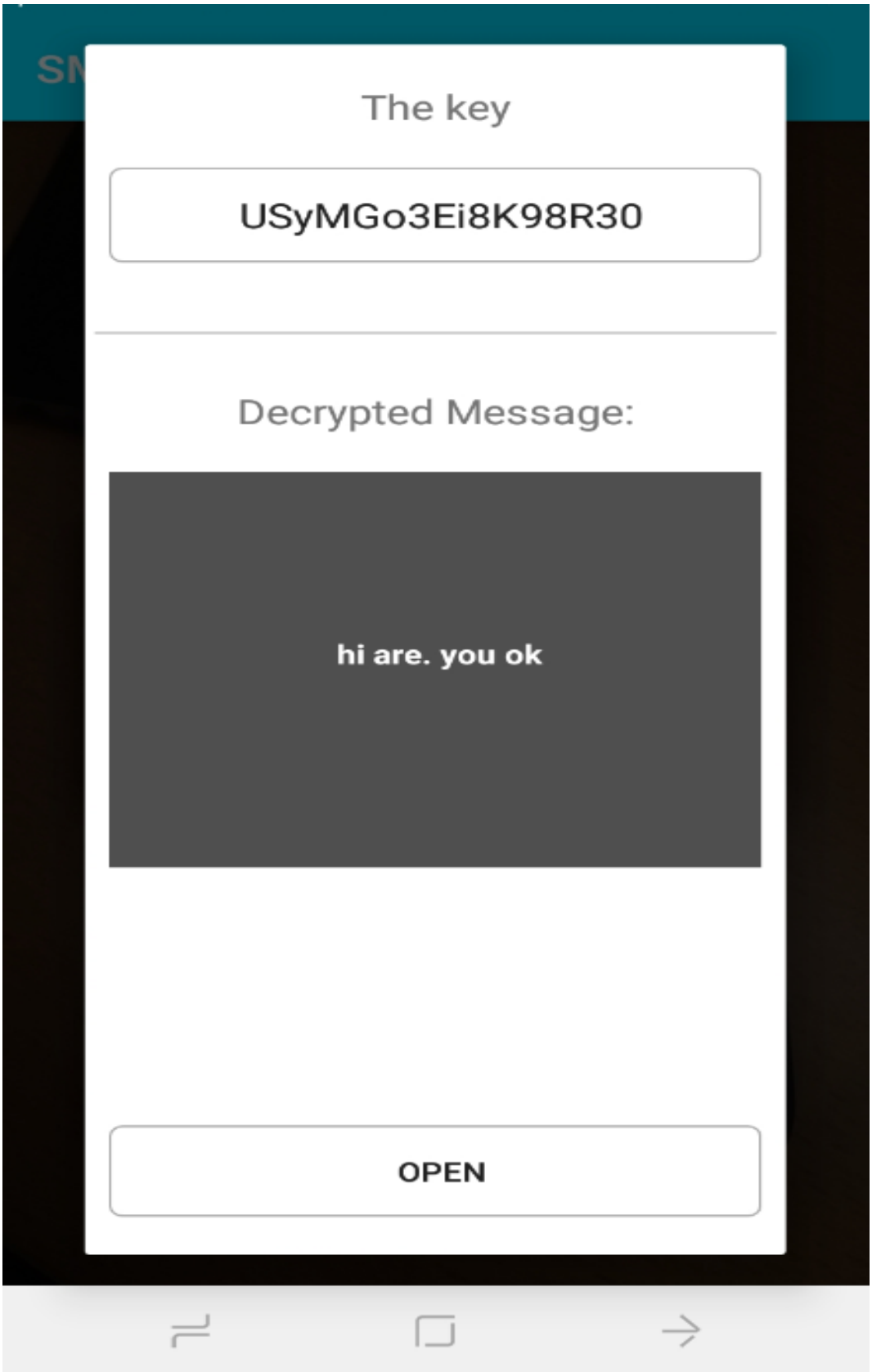

Figure (4.23): *The original message.*

4.6 Reviewing objectives:

After reviewing the objectives mentioned in chapter (1), which is to exploit the features of the mobile device to the user, enter them as a mechanism to authenticate the user on the computer system, to provide more security and strong authentication of the approach, we note that the application has fulfilled the objectives, to provide multiple levels of protection and require effective authentication and a high level of precision.

4.7 Research limitations:

- If there is no connection between the mobile device and the server, the application hangs up after the allotted time has elapsed.
- If a user can find out another user's full information (Username, Password, User ID and padding), they can take advantage of that information and login to the system from any

4.8 Analysis of results:

This application enhanced the security of SMS sent using Android mobile phones. It stands different from other systems as it runs in the mobile phone and does not require any additional encryption devices. The results showed the application is suitable and easy to implement in mobile devices. It also works in any Android environment and supports languages; these features make this application unique one. It is also more secure because if an attack happens to the application, the attacker will receive unreadable messages.

Chapter Five

Conclusions and Recommendation

CAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions:

The application of SMS Encryption using AES algorithm on Android application has been designed and implemented. The application runs in the mobile phone and does not require any additional encryption devices. The results showed that it is suitable and easy to implement in mobile devices for the proposed scheme. With the increased use of SMS for communication and information exchange, care should be taken when sensitive information is transmitted using SMS. Users should be aware that SMS messages might be subject to interception. Solutions such as encrypted SMS should be considered if there is a need to send sensitive information via SMS.

5.2Recommendations:

To improve the application, the following future work is recommended:

1. Enhance the application to encrypted image, audio, video and other information formats.
2. Enhance the application to have a data base to store massages.
3. Find other approaches to increase the level of security and reduce Complexity.

References:

[1] JalaluddinKhana, HaiderAbbasa,b*, Jalal Al-Muhtadia on Mobile User's Data Privacy Threats and DefenseMechanisms, International Workshop on Cyber Security and Digital Investigation (CSDI 2015)

[2] Yan Q., Y. Li, T. Li, and R. Deng, "Insights into Malware: Detection and Prevention on Mobile Phones," in Security Technology, D.S´lzak, T.-h. Kim, W.-C. Fang, and K. P. Arnett, Eds. Springer Berlin Heidelberg, 2009, vol. 58, ch. 30, pp. 242–249

[3] Rohan RayarikarB.E in Computer Engineering SMS Encryption using AES Algorithm on Android ,International Journal of Computer Applications (0975 – 8887) Volume 50– No.19, July 2012.

[4] Namrata A. Kale, Prof. S. B. Natikar, Priyanka D. Navgire, Secured Mobile Messaging for Android application by using 3D-AES, PGP and Stegnography International Journal of Innovative Research in Computer and Communication Engineering/2015

[5] BehnamBazli, Mustafa Anil Tuncel and David Llewellyn-Jones , DATA ENCRYPTION USING BIO MOLECULARINFORMATION ,International Journal on Cryptography and Information Security (IJCIS), Vol. 4, No. 3, September 2014

[6] CătălinBoja, Paul Pocatilu, AlinZamfiroiu, Data Security in M-Learning Messaging Services, INTERNATIONAL JOURNAL OF COMPUTERS AND COMMUNICATIONS(2011)

[7] Raluca Ada Popa, Emily Stark,† Jonas Helfer, Steven Valdez,NickolaiZeldovich, M. FransKaashoek, and HariBalakrishnan , Building web applications on top of encrypted data using Mylar , MIT CSAIL and †Meteor Development Group/, 2016 ,.

[8] /Ako Muhammad Abdullah,RozaHikmat Hama Aziz MSc , New Approaches to Encrypt and Decrypt Data in Image using Cryptography and Steganography Algorithm , International Journal of Computer Applications (0975 – 8887) Volume 143 – No.4, June 2016. Computer Science University of Sulaimani Kurdistan Region-Iraq, 2016 .

[9] Ammar H. Ali, 2Ali MakkiSagheer , Design of a secure android chatting application using end to end encryption , JOURNAL OF SOFTWARE ENGINEERING & INTELLIGENT SYSTEMS ISSN 2518-8739 30th April 2017 .

[10] J.Daemen and V.Rijmen, AES Proposal: Rijndael, NIST"s AES home page, http ://www:nist:gov/aes."Announcing the Advanced Encryption Standard (AES)",Federal Information Processing Standards Publication 197, November 2001 (1).

[11] PriyankaPimpale, RohanRayarikar and SanketUpadhyay, “Modifications to AES Algorithm for Complex Encryption”, IJCSNS International Journal of Computer Science and Network Security, VOL.11 No.10, October 2011.

[12] Jalaluddin Khana , Haider Abbasa,b*, Jalal Al-Muhtadia , ,Survey on Mobile User's Data Privacy Threats and DefenseMechanisms , International Workshop on Cyber Security and Digital Investigation (CSDI 2015) .

[13] Muhammad Noman Riaz a, Adeel Ikram b , Development of a Secure SMS Application using Advanced Encryption Standard (AES) on Android Platform , Mathematical Sciences and Computing, 08 April 2018

[14]https://en.wikipedia.org/wiki/Android_software_development//

[15] Varsha S. Bari1,Nileema R. Ghuge2,Chaitali C. Wagh3,Sayali R. Sonawane 4,Mr.M.B. Gawali5 , SMS ENCRYPTION ON ANDROID MESSAGE APPLICATION,1 Student, Information Technology, Sanjivani College of Engineering Kopargaon , Maharashtra, India2016

[16] Pejman Dashtinejad, Professor Sead Muftic ,Security System for Mobile Messaging Applications,Master of Scienc Department of ICT KTH University SE-100 44 Stockholm, Sweden TRITA–ICT–EX-2015:2 , 8 January 2015.

[17] https://en.wikipedia.org/wiki/Android_software_development//

[18]http://www.dre.vanderbilt.edu/~schmidt/android/android 4.0/out/target/common/docs/doc-comment-check/sdk/eclipse-adt.html.//

[19]https://en.wikipedia.org/wiki/Android_Studio//

Appendix

Appendix A. Decryption Algorithm

```
// decryption function

public static byte[] decryptSMS(String secretKeyString, byte[] encryptedMsg)
throws Exception {
// generate AES secret key from the user input secret key
Keykey = generateKey(secretKeyString);
// get the cipher algorithm for AES
Cipher c = Cipher.getInstance("AES");
// specify the decryption mode
c.init(Cipher.DECRYPT_MODE, key);
// decrypt the message
byte[] decValue = c.doFinal(encryptedMsg);
    return decValue;
}
```

```
private static Key generateKey(String secretKeyString) throws Exception {

// generate AES secret key from a String

Key key = new SecretKeySpec(secretKeyString.getBytes(), "AES");

    return key;
}
```

```
// user input the AES secret key

String secretKeyString = sender.getText().toString();


//key length should be 16 characters as defined by AES-128-bit

if (secretKeyString.length() >0

&&secretKeyString.length() == 16) {

try {
```

```
// convert the encrypted String message body to a byte

        // array

byte[] msg = hex2byte(msgContent.getBytes());


// decrypt the byte array

byte[] resulti = decryptSMS(secretKeyString, msg);


// set the text view for the decrypted message

decrypt.setText(new String(resulti));

//Toast.makeText(this, ""+resulti, Toast.LENGTH_SHORT).show();


Log.d("hhi", String.valueOf(resulti));
```

```
// encrypt the message and send when click Send button

send.setOnClickListener(new View.OnClickListener() {

public void onClick(View v) {

        String recNumString = recNum.getText().toString();

        if (sharedpreferences.contains(Names)) {
secretKey.setText(sharedpreferences.getString(Names, ""));
Log.d("sosa", sharedpreferences.getString(Names, ""));
}
        String secretKeyString = x2;

String msgContentString = msgContent.getText().toString();
Log.d("sarr", x2);

// check for the validity of the user input

        // key length should be 16 characters as defined by AES-128-bit

if (recNumString.length() >0 &&msgContentString.length() >0) {


// encrypt the message

byte[] encryptedMsg = encryptSMS(secretKeyString,
```

```
msgContentString);

// convert the byte array to hex format in order for

            // transmission

String msgString = byte2hex(encryptedMsg);

// send the message through SMS

sendSMS(recNumString, msgString);

Toast.makeText(MainActivity.this, "Message sent ", Toast.LENGTH_SHORT).show();
// finish

            //finish();

} else

Toast.makeText(

getBaseContext(),

"Please enter phone number, secret key and the message. Secret key must be 16 characters!",

Toast.LENGTH_SHORT).show();
}

   });

}
```

Appendix B. Encrypted massage Transmission

```
public static void sendSMS(String recNumString, String encryptedMsg) {
try {

// get a SmsManager
SmsManagersmsManager = SmsManager.getDefault();

// Message may exceed 160 characters
     // need to divide the message into multiples
ArrayList<String> parts = smsManager.divideMessage(encryptedMsg);
smsManager.sendMultipartTextMessage(recNumString, null, parts,
          null, null);

} catch (Exception e) {
e.printStackTrace();
}
```

Appendix C. key Generaten

```
private static Key generateKey(String secretKeyString) throws Exception {
// generate secret key from string
Key key = new SecretKeySpec(secretKeyString.getBytes(), "AES");
   return key;
}
```