

Bucharest University of Economic Studies

Faculty of Cybernetics, Statistics and Economic Informatics

IT&C Security Master Program

Steganography Mobile Application

Dissertation thesis

Coordinator

Conf. univ. dr. Doinea Mihai

Master student

Ionescu Radu Ștefan

Bucharest 2020

Statement regarding the originality of the content

I hereby declare that the results presented in this paper are entirely the result of my own creation unless reference is made to the results of other authors. I confirm that any material used from other sources (magazines, books, articles, and Internet sites) is clearly referenced in the paper and is indicated in the bibliographic reference list.

Table of Contents

[1. Introduction 4](#_Toc42459721)

[2. General presentation 6](#_Toc42459722)

[2.1. Mobile application 6](#_Toc42459723)

[3. Informatic system requirements and analysis 9](#_Toc42459724)

[3.1. System requirements 9](#_Toc42459725)

[3.2. Use cases 10](#_Toc42459726)

[3.3. System analysis 12](#_Toc42459727)

[3.4. Static analysis 12](#_Toc42459728)

[3.5. Dynamic analysis 13](#_Toc42459729)

[4. Informatic system design 14](#_Toc42459730)

[4.1. Detailed class diagram 14](#_Toc42459731)

[4.2. Database design 14](#_Toc42459732)

[4.3. User interface design 14](#_Toc42459733)

[4.4. Component diagram 14](#_Toc42459734)

[4.5. Deployment diagram 14](#_Toc42459735)

[5. Application implementation 15](#_Toc42459736)

[5.1. Steganographic algorithms 15](#_Toc42459737)

[6. Application usage 16](#_Toc42459738)

[7. Conclusions and recommendations 17](#_Toc42459739)

[8. Bibliography 18](#_Toc42459740)

# Introduction

For as long as we have known ourselves as humans, besides other problems such as food, water or shelter, we have also had the difficulty of communicating sensitive information. While this dilemma existed in our minds since the birth of humankind, rudimentary solutions started to be implemented back in the ancient times.

Nowadays, in an era of information technology, where virtually anybody owns some kind of device connected to the internet, which provides access to lots of data, digital steganography is the modern way of concealing messages or other delicate information. As opposed to ancient steganography techniques, which were implemented in a physical manner, using special ink, writing the message in an unobservable area of a letter or using certain rules for extracting the concealed information from an otherwise normal looking text, the modern implementations makes use of the digital form in which information circulates now.

The term steganography comes from the Latin word “steganographia” which originated from the Greek words “steganos”, meaning concealed and “graphia”, meaning writing, so steganography describes, generally, any kind of concealed writing.

Digital steganography is the process of embedding sensitive pieces of data into another computer file in such a manner that human perception cannot detect the difference between the original file and the one containing added information. For the purpose of this thesis, the confidential data will be referred to as the secret message and the container which embeds it will be called the cover file. Generally, both the secret message and the cover file can be of any computer file type. For example, we can hide: a plain text file into an image file, a picture into a sound recording or an image into a text file. However, it is not always possible to do this due to the sizes of the files, so hiding some text into an image file may be far easier than the other way around because, generally, an image file is bigger in size than a text file, but more important, a text file doesn’t contain redundant information, in comparison with a picture. For this reason, this thesis will describe how to embed secret text messages in image and sound recording files. This computer mediums are generally large and redundant enough that we can alter them, by embedding the secret information inside, but to the human eye or ear it will appear the same. This is a clever way to send sensitive information because, while cryptography ensures the message is not readable, steganography does not reveal the existence of the message at all.

Additionally, encryption will be used in order to ensure better security for the message, by encrypting it before embedding. Cryptography is different than steganography because, while encrypting a message ensures its transformation into a form that an eavesdropper would not understand, steganography tries to hide the existence of a hidden message at all.

Other similar processes tightly coupled with steganography are fingerprinting and digital watermarking. A fingerprinting algorithm is used to generate a unique mark for a piece of data and embed it into that specific file. This is very useful when you want to supply some files and protect them from ongoing distribution. Watermarking also embeds a mark of the files with the purpose of signifying ownership. As opposed to steganography, in fingerprinting and watermarking the existence of the embedded data is publicly known, whereas steganography tries to completely hide that there is any information hidden inside. While a successful attack concerning a watermarking or fingerprinting algorithm consists of removing the watermark or fingerprint, basically removing the ownership protection, an attack on a steganographic system should detect and eventually extract the hidden data.

LSB or Least Significant Bit steganography is a technique which implies using the redundancy of the information as a place to store the data to be hidden. This is the reason multimedia files are so popular with steganography. They contain a fair amount of redundant information, whether we discuss about image, audio or video files, which makes the perfect home for secret messages. On the other hand, from a statistical point of view, if the message is embedded directly, as it is, into a cover file, it may be easily discovered, because altering the original file this way changes its statistical properties, if the secret message is large enough.

This is another good reason to also use cryptography when applying this kind of technique. Besides securing the content of the message, through encryption, the original message is transformed in a more random looking form, that of a ciphertext. This way, the statistical impact of the embedding process is drastically reduced because the randomness of the least significant bits is not altered heavily. Also, the ciphertext should be as evenly as possible distributed through the cover file, for a better security over statistical methods of examination.

# General presentation

This thesis describes the development of an Android mobile application used for peer to peer communication, or more informal, chatting. The application uses Google Firebase as a backend service for multiple purposes like authentication, real-time database and persistent storage. The application should allow and facilitate users to create accounts and send and receive different kinds of messages.

The main information the application needs and uses is made up of the user details and the messages exchanged between them. This information is taken from different forms filled in by the users.

# Mobile application

Users can communicate with each other using the mobile application, which should be up and running at any time. For a good user experience, the required data will be filled in by the users in simple, intuitive forms and the buttons and different options will be named as evocative as possible. The information presented to the users will be displayed in frames whose elements will be logically placed, for the ease of understanding.

A user must register to use the StegLock application by filling the registration form or by using a personal social account, like Facebook or Google. By using a social account, all the required data about the user will be taken from there, with his permission. If the client chooses to complete the registration form, he needs to enter his personal data, which is the full name, a nickname, an email address, a profile photo and to create a password. Filled in data is verified in real time and the form doesn’t allow the creation of accounts with flawed data, indicating which of the fields contain wrong information and what are the rules for each.

After registration, the client needs to log in to use the application, by entering the email and the password or by using his Facebook or Google social account. For a good user experience, the application allows persistent user authentication. This means that, unless the user specifically wants to log out, he will remain authenticated even if the mobile device is shut down or restarted. Once logged in, the main application menu is presented to the user. This menu allows has four options. The options are chatting, displaying the user information, displaying application information and logging out.

To start chatting, after selecting the chat option from the main menu, the user is presented the chat frame, split into two tabs, the users tab and the chats tab. The users tab displays a list of all of the registered users and at the top a text box is used to search for users by their nickname. A tap on a user from the list will open a new conversation with that specific user. The chats tab displays a list of users with whom conversations already exist. For new users the chats tab will be empty until the users receives a message or sends one. A tap on a nickname from the chats tab will open the conversation.

The conversation frame displays the different kinds of messages sent between clients. The messages sent by the current user are displayed on the right side of the screen and the received messages are displayed on the left side. The application allows users to send text messages, photos, images or sound recordings.

In the bottom of the frame a text box is used for the text message and four buttons are used to either send the text message, choose an image, take a photo or record a sound. If the image button is pressed, the photo gallery is opened for the user to select the image he wants to send. After the image is selected, it is displayed in a new form and a text box allows the user to input a secret message to be embedded into the image, then the image can be sent by tapping the send button. Similarly, if the photo button is pressed, the process is the same, but instead of the gallery, the camera of the mobile device is opened for the user to take a photo. If the record a sound button is pressed, a new frame is displayed, containing two buttons, one for start/stop recording and one for playing the recorded sound. Also, like the image/photo frame, this one allows the user to input the secret message to be embedded into the recording.

The conversation frame displays text messages, images, photos and sound recordings. The sound recordings are displayed as play buttons and a press on a recording will begin the playback of the recording. A long press on either of images, photos or audio recordings will show a dialog displaying the secret message embedded into the file.

The user profile information frame, accessible from the main menu, displays all of the information about the client and allows him to edit his profile picture or nickname, by pressing the appropriate buttons. These buttons display a dialog in which the user should input the data to be changed or select an image, in case of profile photo editing. Also, the edited details are validated in real time and flawed information is indicated in the form along with the rules to be followed to fill in correctly.

The application information frame displays a description of the application and the signification of different symbols and buttons used in the application. Also, here can be found a short tutorial on how to use the chatting functionality and how are the secret messages being embedded into the cover files. The steganographic algorithms used to create images or sounds with embedded messages are shortly described here.

# Informatic system requirements and analysis

# System requirements

The functional requirements of the informatic system will be identified and modeled using use case diagrams, which outlines the way in which the system is used, by highlighting the actors and the actions that can be done by them.

Below, in *Figure 3.1*, the general use case diagram is shown along with a text description of each one, elaborating the purpose, the main actor, the preconditions, the basic and alternate flows and the usage frequency of the respective use case.

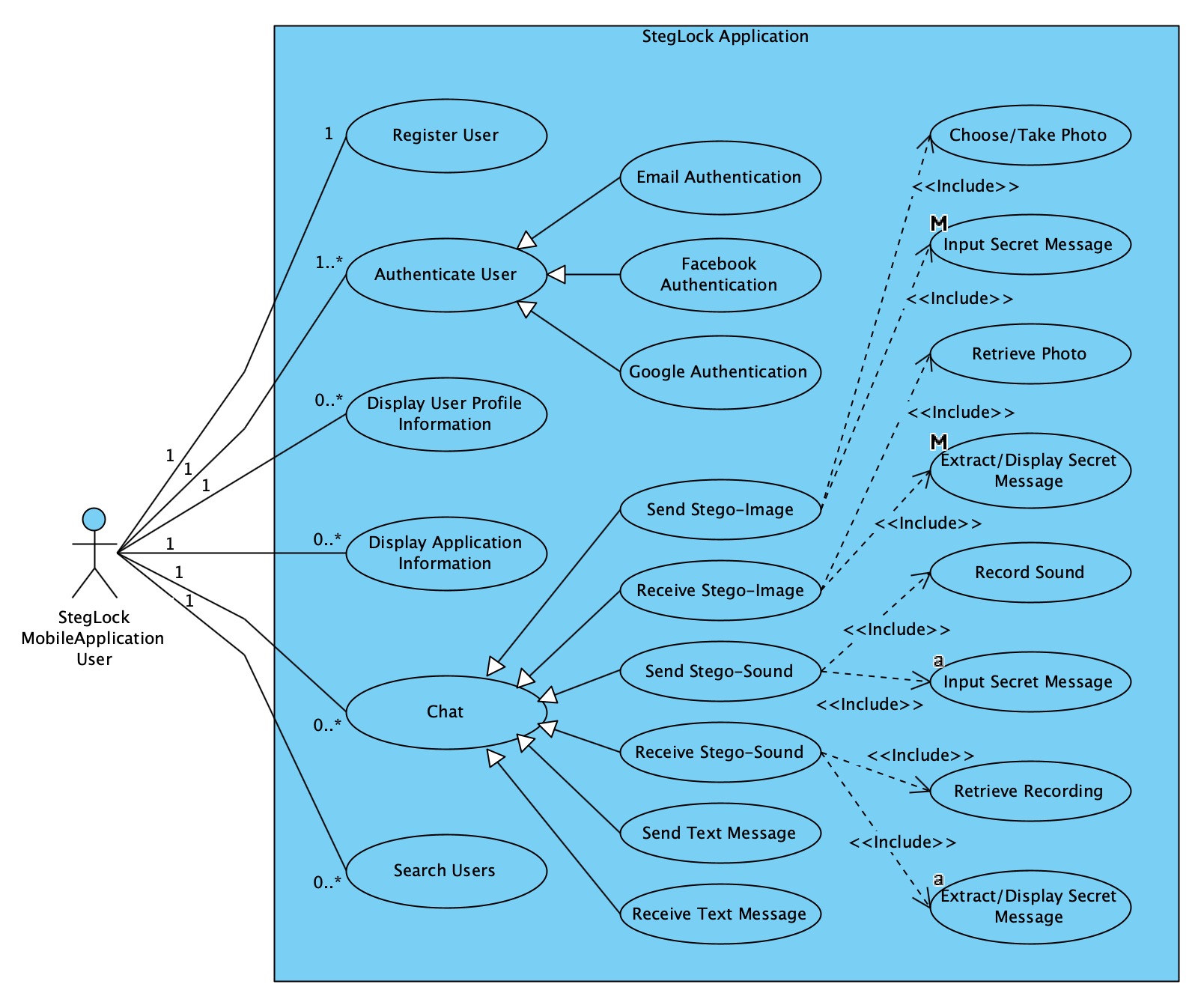


Figure 3.1 - Use case diagram

The above diagram covers all the possible usages of the system. The next chapter explains in detail the functionality of the system, by presenting each of the use cases present in the above diagram and detailing each one’s specificities.

# Use cases

The “Register User” use case has the purpose of creating an account for a user. The main actor, in this case, is the client, who starts this process. The completion of this use case is preconditioned by an internet connection and the client knowing his personal information needed to create the account. The basic flow of this use case is:

1. Choose profile picture
2. Input nickname
3. Input first name
4. Input last name
5. Input email address
6. Input password
7. Input confirmation password
8. Confirm data and create account

Alternatively, the user can cancel the process at any time and no account will be created. The usage frequency of this use case is low, because every client may have to complete this scenario once.

The “Authenticate User” use case has the purpose of signing in a client that has an account and it represents a generalization of the three main authentication methods. The main actor, in this case, is the application user, who initiates this scenario. The completion of this use case is preconditioned by an internet connection and the client knowing his credential information or social accounts credentials needed to sign into the account. The basic flow of this use case is:

1. Input email address
2. Input password
3. Authenticate

Alternatively, the client can use a social platform account, namely a Google or Facebook account, to get authenticated or he can cancel the process at any time and no account will be authenticated. The usage frequency of this use case is moderate to low, because every client may have to complete this scenario once or a few times as he will remain persistently authenticated until he wants to sign out.

The “Display User Profile Information” use case has the purpose of displaying account information to the client. The main actor is the application user, who initiates this scenario. The completion of this use case is preconditioned by an internet connection and the client being signed into his account. The basic flow of this use case is:

1. Select “My Profile” from menu
2. Display account details

The usage frequency of this use case is moderate to low, because every client may check his profile information from time to time, when changes about his information appear.

The “Chat” use case has the purpose of communication between the users and it represents a generalization of the multiple ways in which clients can exchange messages. The main actor, in this case, is the client, who starts this process. The completion of this use case is preconditioned by an internet connection and the client being authenticated. The specific chat use cases are “Send Text Message”, “Receive Text Message”, “Send Stego-Image”, “Receive Stego-Image”, “Send Stego-Sound” and “Receive Stego-Sound”.

The basic flow of the “Send Text Message” use case is:

1. Input text message
2. Send message

The basic flow of the “Receive Text Message” use case is:

1. Retrieve text message
2. Display message

The basic flow of the “Send Stego-Image” use case is:

1. Choose image or take photo
2. Input secret message
3. Embed secret message into image
4. Send image

The basic flow of the “Receive Stego-Image” use case is:

1. Retrieve image
2. Extract secret message from image
3. Display image
4. Display extracted secret message

The basic flow of the “Send Stego-Sound” use case is:

1. Record audio
2. Input secret message
3. Embed secret message into sound recording
4. Send recording

The basic flow of the “Receive Stego-Image” use case is:

1. Retrieve recording
2. Extract secret message from recording
3. Play sound recording
4. Display extracted secret message

Alternatively, the user can cancel the process at any time and no messages will be sent. The usage frequency of this use case is high, because this is the main use case of the system and client may chat frequently.

# System analysis

The analysis of the system implies the examination of the requirements and use cases, both from a static and dynamic point of view, identifying and highlighting the basic concepts the system is working with and also, the existing relationships between them.

# Static analysis

The static analysis of the system implies the development of the class diagram, by identifying the main classes, necessary for the operation of the system. In the analysis stage, a simplified class diagram is developed and later, in the system design stage the detailed class diagram will be developed, by specifying for each class the set of attributes, methods and constraints.

Analyzing the system’s use cases and requirements, two main classes are identified. Those are the “User” class and the “Message” class and in *Figure 3.2 – Simplified class diagram* those are presented, along with the identified relationships between them.

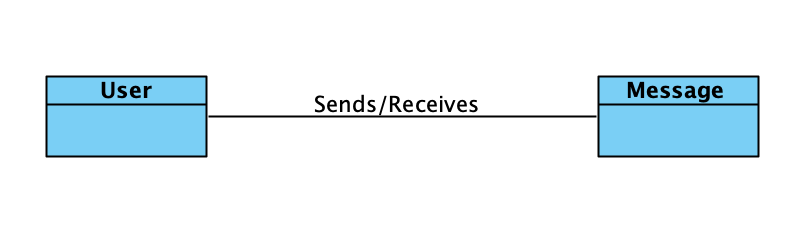


Figure 3.2 – Simplified class diagram

# Dynamic analysis

The dynamic analysis of the informatic system implies observation of the evolution of its components over time, by developing some specific diagrams, namely state machine diagrams, activity diagrams and interaction diagrams.

The state machine diagram highlights the states through which an object or event can be found at certain moments in time and represents the lifecycle of the objects. The development of this kind of diagram also emphasizes the transitions between the states, by identifying the events that causes them.

For an object of User type, four possible states are identified, namely:

* Registered User
* Authenticated User
* Logged Out User
* Deleted User

In *Figure 3.3 – User state machine diagram* the states and transitions between User objects is detailed.

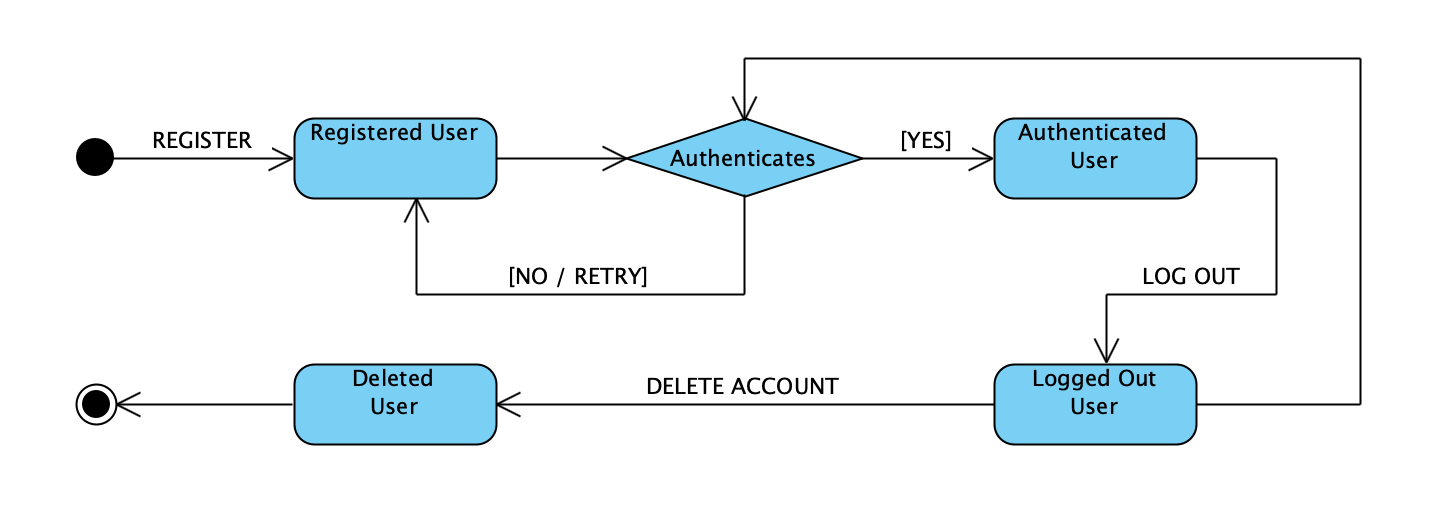


Figure 3.3 - User state machine diagram

For an object of Message type, six possible states are identified, namely:

* New Message
* Text Message
* Stego-Image Message
* Stego-Sound Message
* Sent Message
* Deleted Message

In *Figure 3.4 – Message state machine diagram* the states and transitions between User objects is detailed.

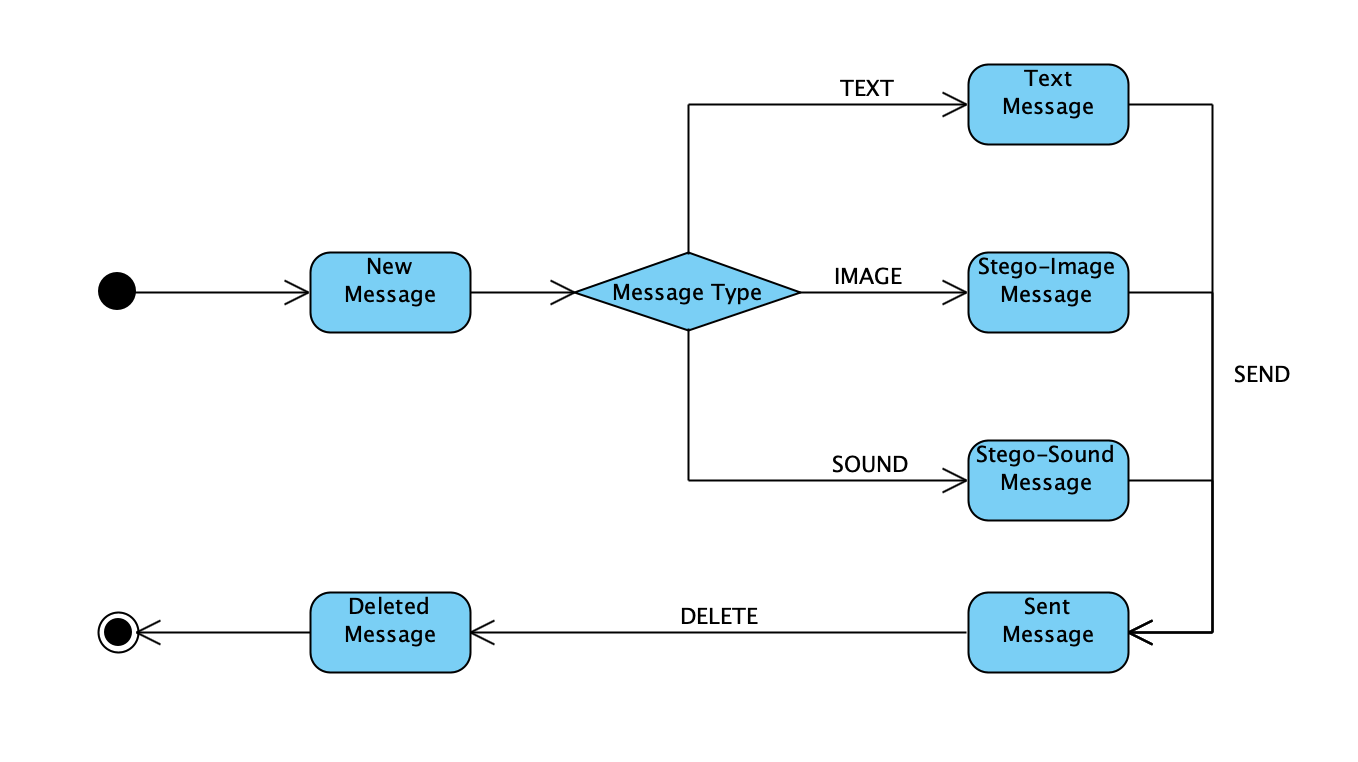


Figure 3.4 - Message state machine diagram

The activity diagram allows the highlighting of the work flows by identifying the action sequences and the results of those, also highlighting the decisions that may appear by executing the respective action.

Next, in *Figure 3.5 – Chat activity diagram*, it is presented the diagram of the chat activity, which is the main one. The diagram is made up of two vertical lanes, named partitions, representing the entities that participate in this activity. Also, there are multiple round boxes positioned inside the partitions, representing the actions to be made in this activity.

Control is passed from an action to another by using control flows or object flows. Control flows are represented by a simple arrow and object flows are represented by arrows along which square boxes are existing, boxes that represent the objects or data transferred between actions. The beginning and the end of the activity is marked by the initial and the final node.

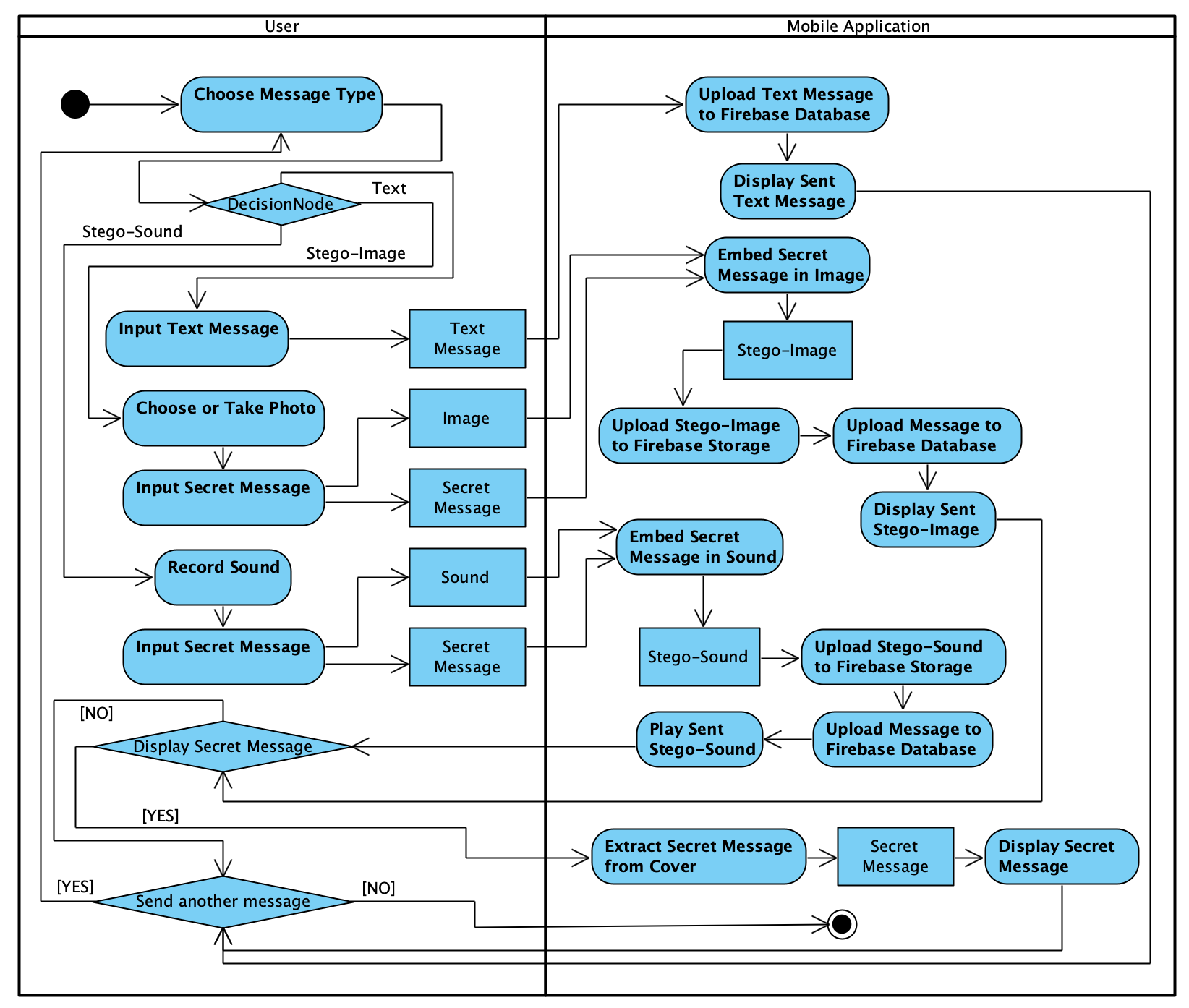


Figure 2 - Chat activity diagram

The interaction diagrams are highlighting the dynamic aspects of the system and are made up of a set of objects and the relationship between them, exposing in the same time their interaction, through the messages exchanged from one to another.

There are two types of interaction diagrams, the sequence diagram and the communication diagram. The two types of diagrams are equivalent from the point of view of their meaning, but both should be developed because the sequence diagram highlights the ordering and interaction between objects in time, while the communication diagram exposes the structure of the objects that are sending and receiving messages.

The sequence diagram is composed of the objects that interact with each other, placed on vertical lanes. Under these objects, their lifeline is represented using a discontinuous line, which signifies the existence of an object, in a certain period of time. Along the lifelines of the objects, tall rectangles mark an action done by the object, but also the start, end and duration of it on the timeline.

The messages sent between objects are represented by horizontal arrows which are crossing the vertical lifelines. Those arrows are connecting two control points of the various objects which communicate or they can point to the same object, signifying self-communication. At the same time, those arrows also indicate the moment in time at which the action is done, through the place they are placed in.

Next, in *Figure 3.6 – Chat sequence diagram*, the diagram is shown, detailing the objects that are participating in the chat action along with the communication that is taking place between them and their evolution in time.

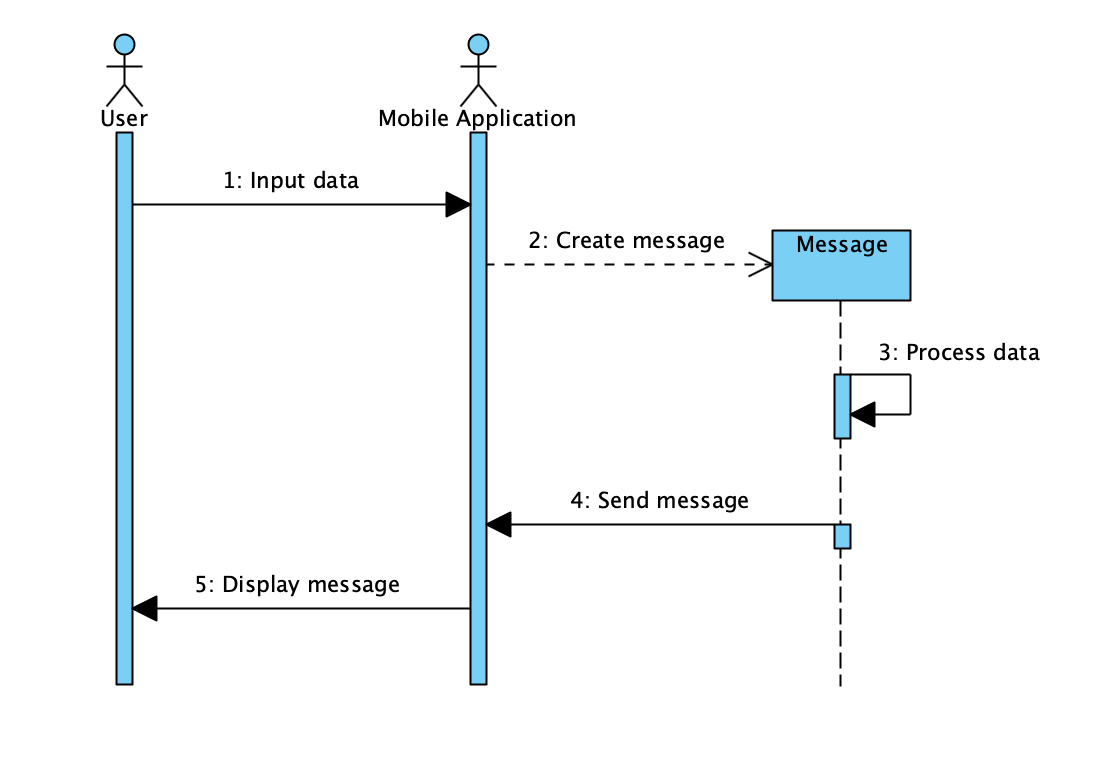


Figure 3.6 - Chat sequence diagram

The communication diagram emphasizes the structure of the objects and actors which are sending and receiving several messages. This diagram is represented like a graph whose nodes represent the participants and whose edges represent the connections between them. Along the edges between objects, there are the sent messages, prefixed by an order number. The direction of the messages is highlighted by using arrows.

In *Figure 3.7 – Chat communication diagram*, the diagram is presented.

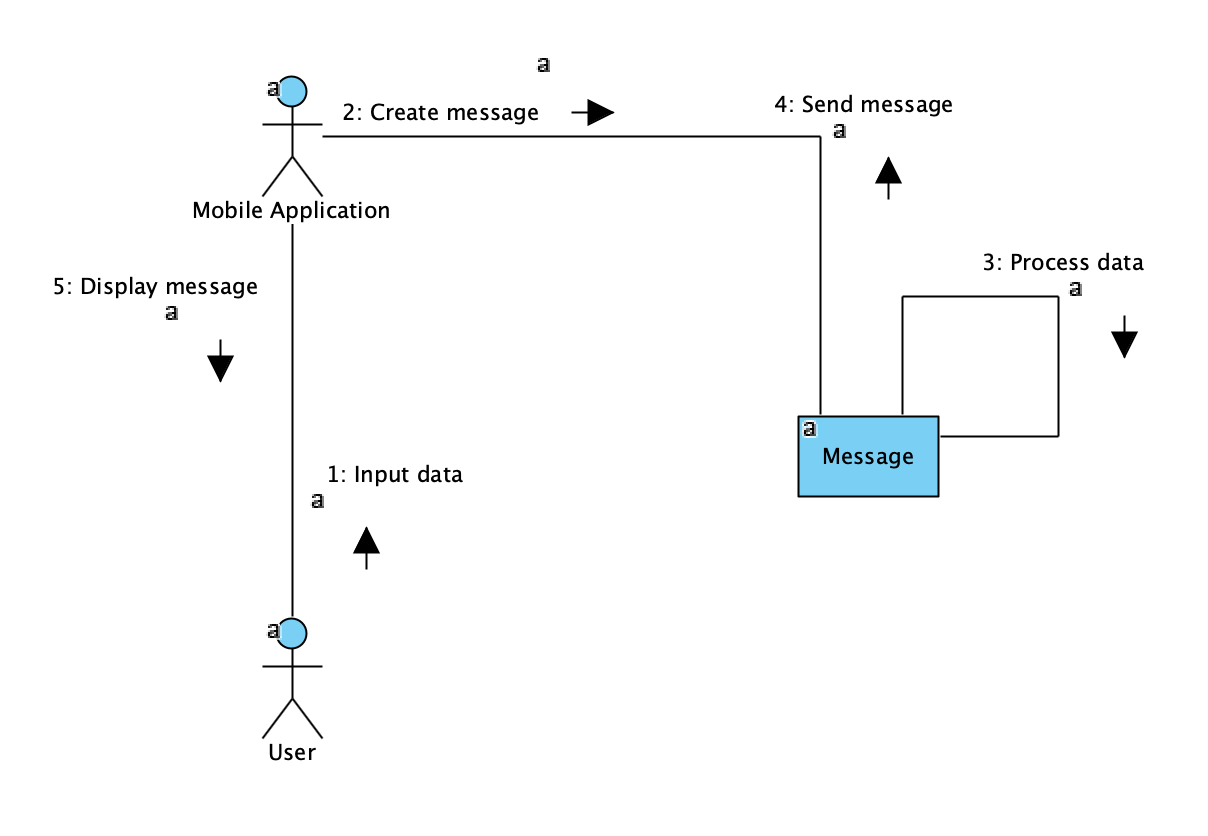


Figure 4.7 - Chat communication diagram

# Informatic system design

# Detailed class diagram

# Database design

# User interface design

# Component diagram

# Deployment diagram

The deployment diagram exposes the structure of the system at the execution moment, by highlighting the various equipment necessary for the operation of the system, along with the connections between them.

The systems necessary components are:

* Mobile Device – the mobile device of the client, on which the StegLock mobile application will run.
* Firebase Development Platform – the development platform from Google which provides necessary services like Authentication, Storage and Database.
* Internet Network – the network which connects all of the equipment.

In *Figure 4.2 – Deployment diagram* this equipment is presented.

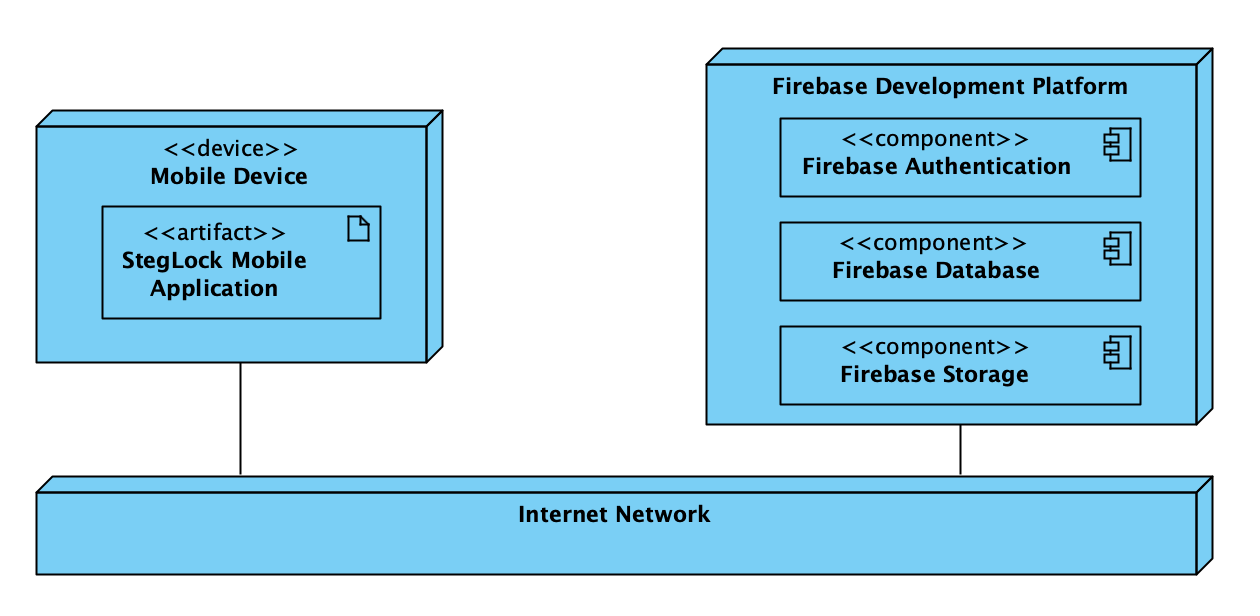


Figure 4.2 - Deployment diagram

Following the system design and analysis of the system requirements the implementation will be done accordingly, as described in all the diagrams presented in the last two chapters. This will be done using an appropriate, high-level programming language, namely Java, suitable for the Android execution environment.

# Application implementation

# Steganographic algorithms

# Application usage

# Conclusions and recommendations

# Bibliography

|  |  |
| --- | --- |
| [1] | R. G. BALDWIN, "Processing Image Pixels Using Java: Controlling Contrast and Brightness," [Online]. Available: https://www.developer.com/java/other/article.php/3441391. [Accessed 12 March 2020]. |
| [2] | R. G. BALDWIN, "Processing Image Pixels using Java, Getting Started," [Online]. Available: https://www.developer.com/java/other/article.php/3403921. [Accessed 12 March 2020]. |
| [3] | S. K. ARORA, "Audio Steganography : The art of hiding secrets within earshot (part 2 of 2)," [Online]. Available: https://medium.com/@sumit.arora/audio-steganography-the-art-of-hiding-secrets-within-earshot-part-2-of-2-c76b1be719b3. [Accessed 11 March 2020]. |
| [4] | S. K. ARORA, "Audio Steganography : The art of hiding secrets within earshot (part 1 of 2)," [Online]. Available: https://medium.com/@sumit.arora/audio-steganography-the-art-of-hiding-secrets-within-earshot-part-1-of-2-6a3bbd706e15. [Accessed 11 March 2020]. |
| [5] | M. S. SHAHREZA, "Stealth Steganography in SMS," *International Conference on Wireless and Optical Communications Networks,* 2006. |
| [6] | M. S. MEGHA, "Methods of Audio Steganography," *International Journal of Engineering and Management Research,* vol. 4, no. 3, pp. 154-156, 2014. |
| [7] | M. ZAMANI, A. MANAF, R. AHMAD, F. JARYANI, H. TAHERDOOST and A. M. ZEKI, "A Secure Audio Steganography Approach," *International Conference for Internet Technology and Secured Transactions,* 2009. |
| [8] | N. HAMID, A. YAHYA, B. AHMAD and O. M. AL-QERSHI, "Image Steganography Techniques: An Overview," *International Journal of Computer Science and Security,* vol. 6, no. 3, pp. 168-187, 2012. |
| [9] | R. CHANDRAMOULI and N. MEMON, "Analysis of LSB based image steganography techniques," *Proceedings 2001 International Conference on Image Processing,* vol. 3, pp. 1019-1022, 2001. |
| [10] | N. F. JOHNSON, Z. DURIC and S. JAJODIA, Information Hiding: Steganography and Watermarking - Attacks and Countermeasures, Boston, MA: Springer, 2001. |
| [11] | C. MAITI, D. BAKSI, I. ZAMIDER, P. GORAI and D. R. KISKU, "Data Hiding in Images Using Some Efficient Steganography Techniques," *Communications in Computer and Information Science,* vol. 260, pp. 1-9, 2011. |
| [12] | P. POCATILU, I. IVAN, A. VIȘOIU, F. ALECU, A. ZAMFIROIU and B. IANCU, Programarea Aplicațiilor Android, Bucharest: Editura ASE, 2015. |
| [13] | N. PROVOS and P. HONEYMAN, "Hide and Seek: An Introduction to Steganography," *IEEE Security and Privacy Journal,* vol. 1, pp. 32-44, 2011. |
| [14] | M. FORTRINI, "Steganography and digital watermarking: A global view.," 2011. |
| [15] | J. H. P. ELOFF, T. MORKEL and M. S. OLIVIER, "An Overview Of Image Steganography," *Proceedings of the Fifth Annual Information Security South Africa Conference,* 2005. |
| [16] | "Documentation | Android Developers," [Online]. Available: https://developer.android.com/docs. [Accessed 10 March 2020]. |
| [17] | "Steganography," [Online]. Available: https://en.wikipedia.org/wiki/Steganography. [Accessed 9 March 2020]. |
| [18] | "What is Steganography? Webopedia Definition," [Online]. Available: https://www.webopedia.com/TERM/S/steganography.html. [Accessed 25 April 2020]. |
| [19] | N. EL-EMAM, "Hiding a Large Amount of Data with High Security Using Steganography Algorithm," *Journal of Computer Science,* vol. 3, pp. 223-232, 2007. |
| [20] | T. Morkel, J. Eloff and M. Olivier, "An Overview Of Image Steganography," *Proceedings of the Fifth Annual Information Security South Africa Conference,* 2005. |
| [21] | "Google Firebase Documentation," Google, [Online]. Available: https://firebase.google.com/docs. [Accessed 21 March 2020]. |
| [22] | R. G. BALDWIN, "Steganography 101 using Java," [Online]. Available: https://www.developer.com/java/ent/article.php/10933\_3530866\_2/Steganography-101-using-Java.htm. [Accessed 12 March 2020]. |