# **CHAPTER 4**

**SYSTEM IMPLEMENTATION**

## **4.1 Introduction**

In this chapter, the tools used to design and implement the new System is documented. It includes choice of tools, choice of programming language, as well as hardware and software requirements.

It also includes User documentation, system testing approach as well as other implementation details employed in the system development.

## **4.1 System Implementation Tools**

## **4.1.1 Built Tool - Gradle**

Gradle is an open-source build automation system that builds upon the concepts of Apache Ant and Apache Maven and introduces a Groovy-based domain-specific language (DSL) instead of the XML form used by Apache Maven for declaring the project configuration. Gradle uses a directed acyclic graph ("DAG") to determine the order in which tasks can be run.

Gradle was designed for multi-project builds, which can grow to be quite large. It supports incremental builds by intelligently determining which parts of the build tree are up to date; any task dependent only on those parts does not need to be re-executed.

## **4.1.2 Repository Platform -GitHub**

GitHub is a web-based hosting service for version control. It is used by several developers around the world to collaborate together on projects and make contributions. GitHub provides support for bug tracking, access control and task management. Developers can do code review and refactoring, contribute on other projects as well as manage different versions of their projects.

## **4.1.3 IDE – Android Studio 3**

## Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It provides the fastest tools for building apps on every type of Android so it is most appropriate for this project.

## **4.1.4 Version Control - Git**

Git is a distributed version-control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed, data integrity, and support for distributed, non-linear workflows.

## **4.2 Choice of Programming Language**

**Java**

Java is a general-purpose computer-programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. The platform for app development in Android is Java. This means that you use the **Java** library and code the applications in Javaprogramming language. The most widely used programming language for android application development is Java.

**Android Framework**

The androidframework is the set of API's that allow developers to quickly and easily write apps for android phones. It consists of tools for designing UIs like buttons, text fields, image panes, and system tools like intents (for starting other apps/activities or opening files), phone controls, media players, etc. It makes it easier to write codes in android studio and develop scalable android applications.

**4.3 System Requirements**

The System requirement describes the important software and hardware needed for the system to be successfully deployed and maintained. These requirements are essential for the application to work efficiently.

## **4.3.1 Hardware Requirements**

For the system to work efficiently, the following hardware is recommended as the minimum hardware requirements for the developed systems:

1. Minimum of 1.2Ghz Processor
2. 2gb RAM
3. Android Device
4. 850MB memory space

## **4.3.2 Software Requirements**

For the system to work efficiently, the following hardware is recommended as the minimum software requirements for the developed systems:

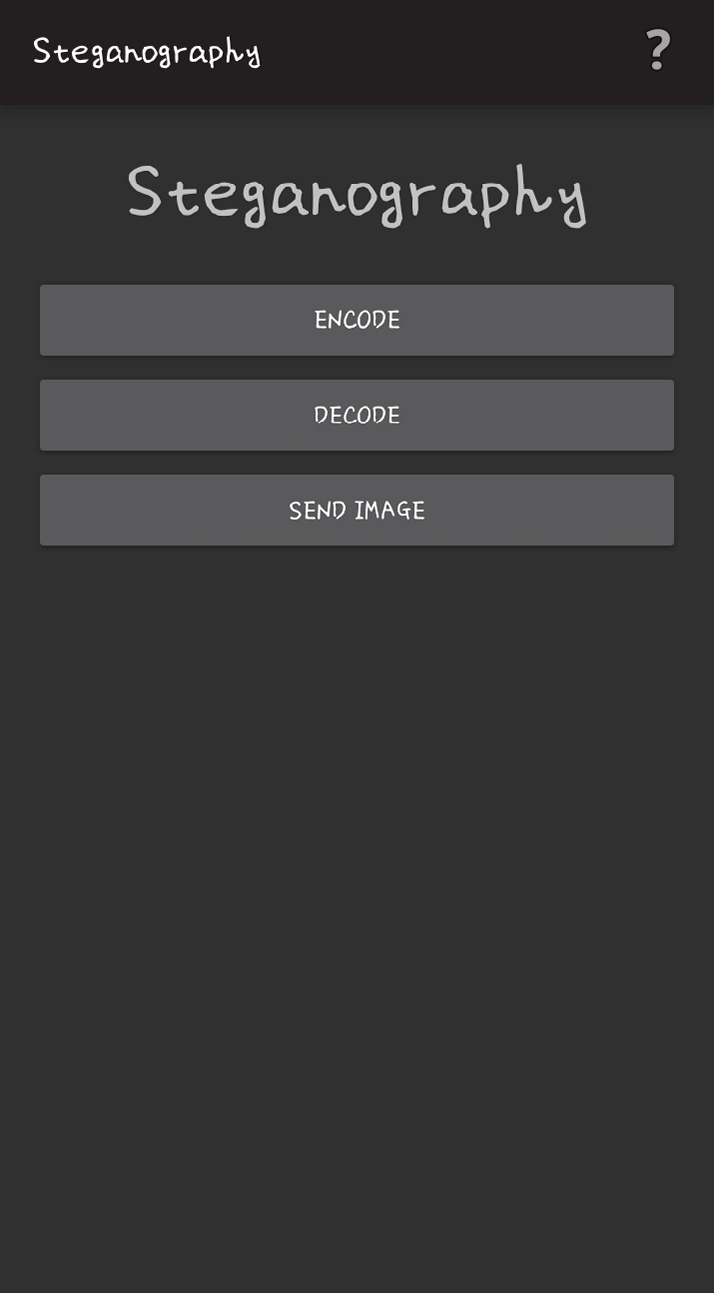
1. Android Jellybean Operating System (4.1-4.3)
2. Java Virtual Machine

## **4.4 System Interface**

## **4.4.1 App Splash Screen Page**

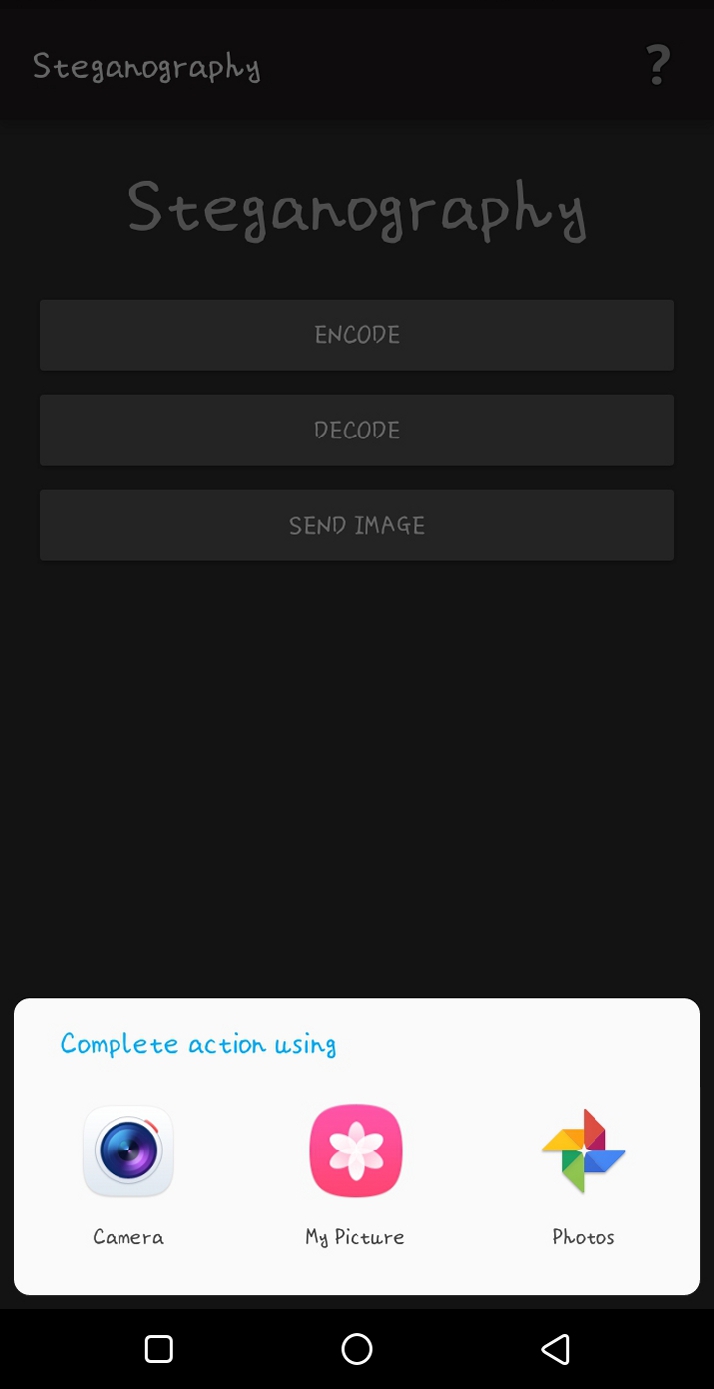
The splash screen is a form of landing page for the application. It has a simple and easy to use interface. This screen contains the basic menu items of the functionalities of the app.

1. ENCODE: this menu when clicked accepts an image, requests for encoded text and encode the text into the image.
2. DECODE: this menu when clicked accepts an image, decodes the image and reveals the encoded text.
3. SAVE IMAGE: this menu when clicked saves encoded and decoded image into phone gallery.

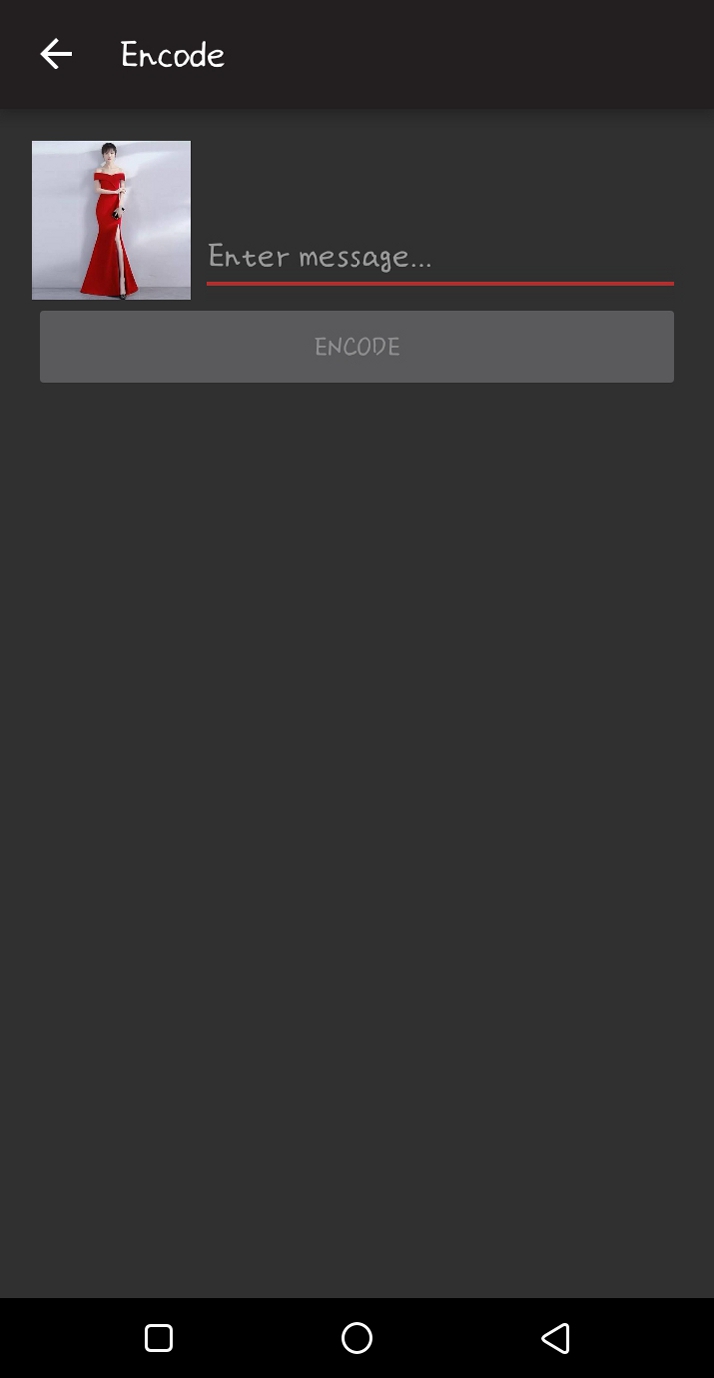


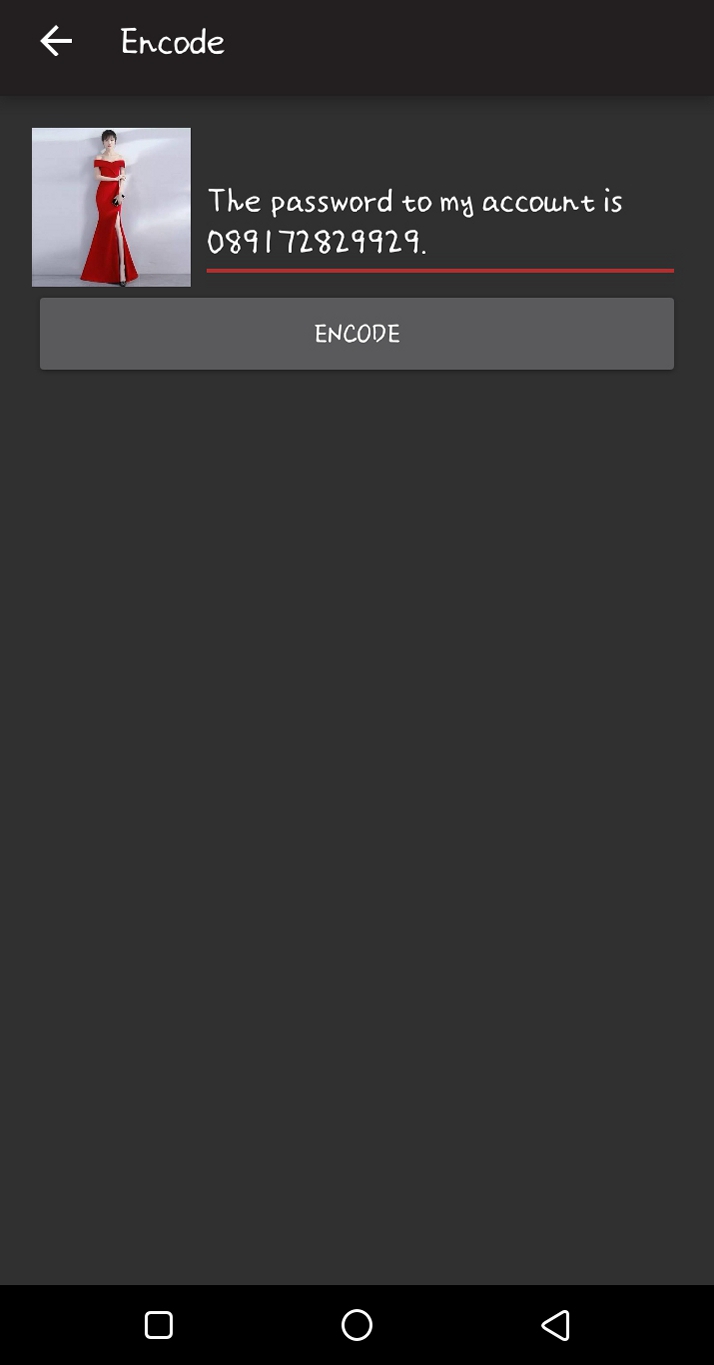
## **4.4.2 Encode Page**

Clicking on the encode menu brings a pop up to select a picture folder for encoded image. Users can choose to take pictures directly from camera or select from the phone’s picture folder.



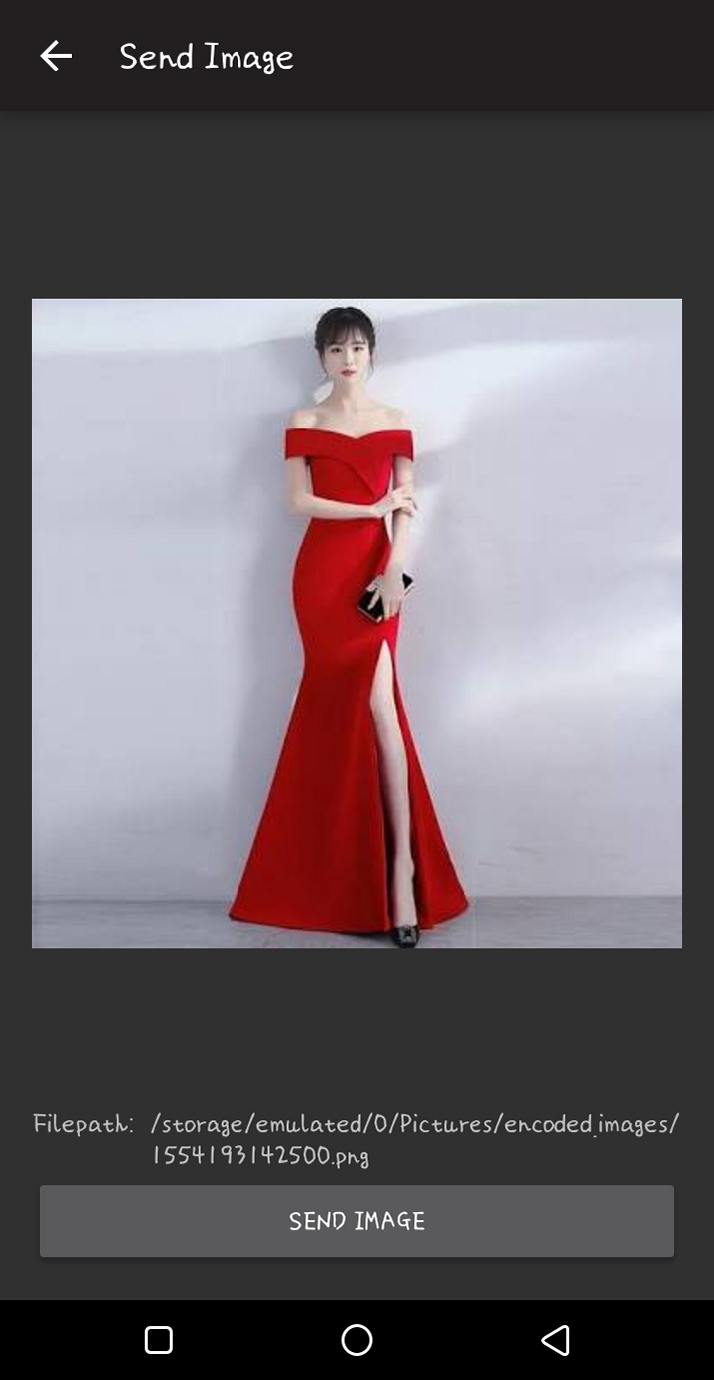
After selecting a picture option and selecting the image to encode, users can input the text to be encoded in the image and then encode the text into the image.





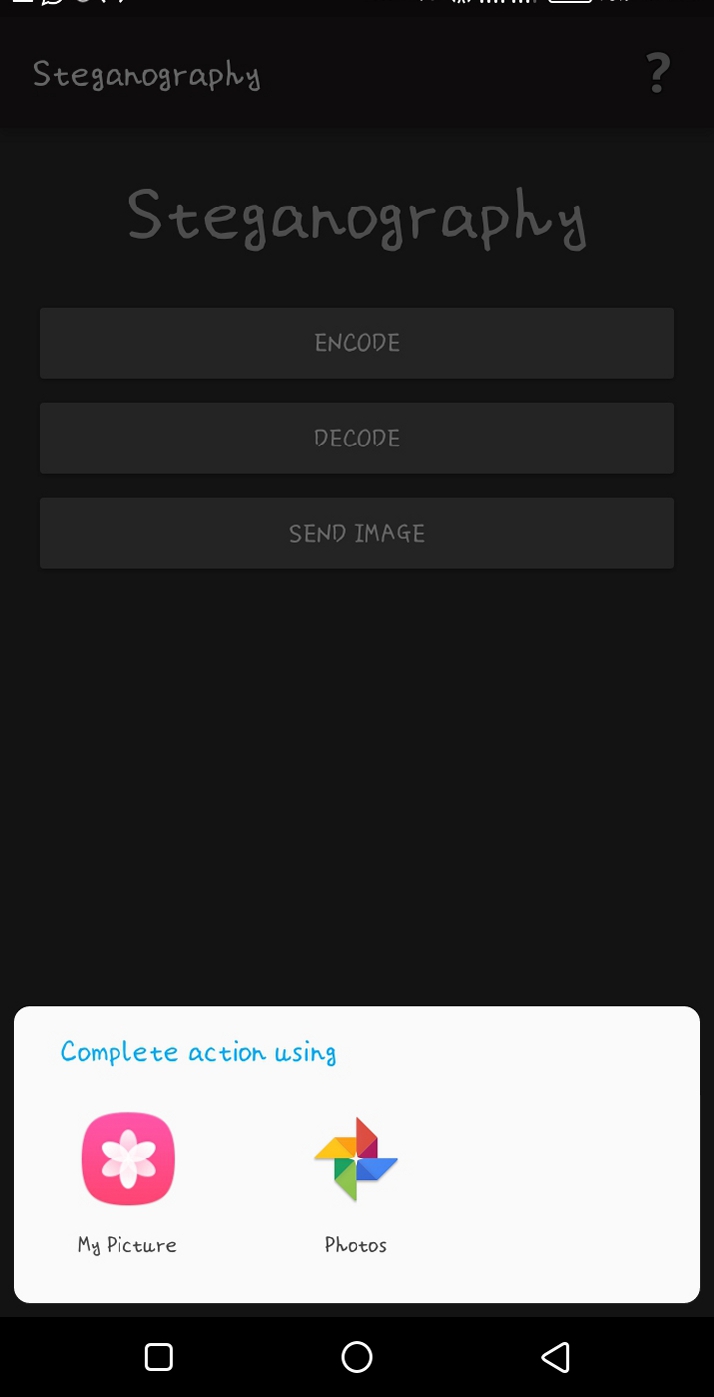
## **4.4.3 Encoded Image page**

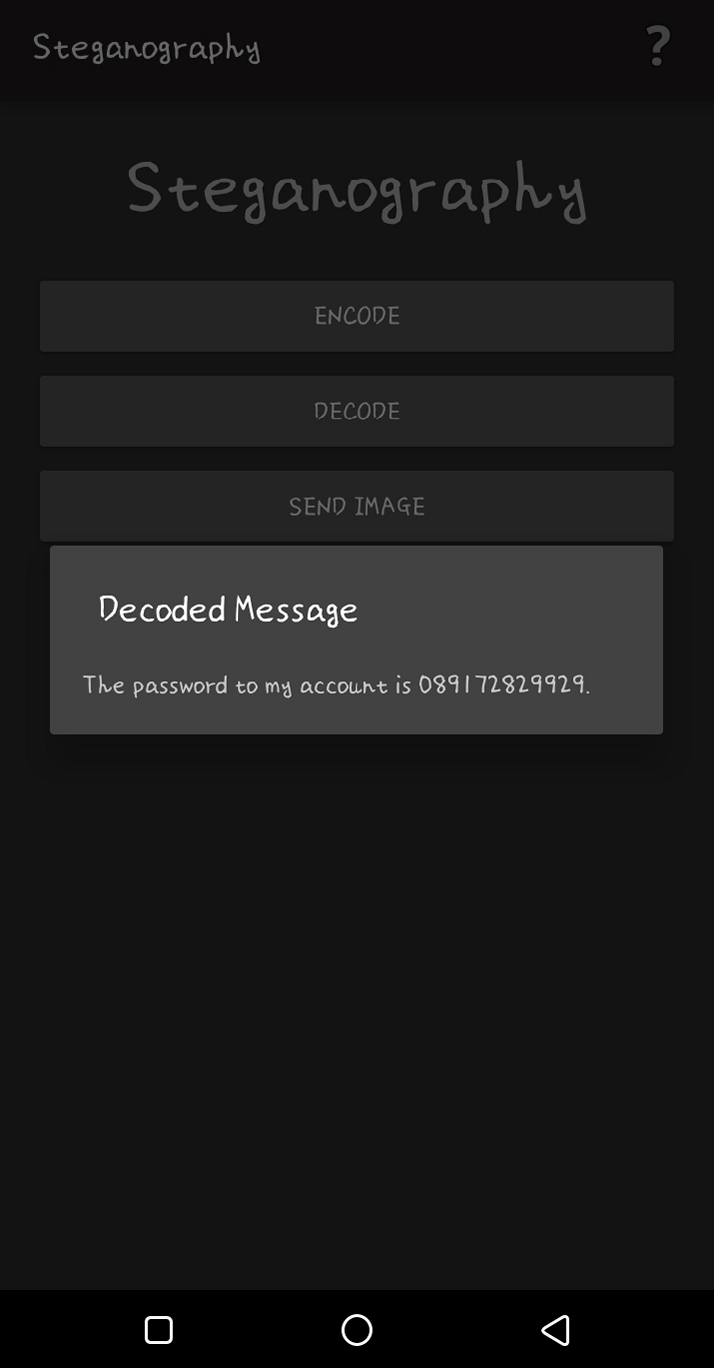
After encoding the image, the image is displayed in a new page where it can be saved in a gallery or sent via different social platforms.



## **4.4.4 Decode Page**

From this page, Users can select the decode menu and a menu pops up with options to select image to be decoded. After selecting the image, the text encoded in the image is decoded and displayed.





## **System Testing**

The system was tested in two phases, in order to check that it conforms to requirements and it performs all functionalities. The testing done are Unit and Integration testing.

## **4.5.1 Unit Testing**

Unit testing also known as testing in small is the testing of individual components and modules in the system. This is done to ensure that the modules can function independently and without error. For this system, the Encode and the decode components were individually tested for errors and bugs. Different pictures and texts were inputted as test cases in order to detect error or bugs in the system.

## **Integration Testing**

Integration testing involves testing all the modules together and integrating the components, also called testing as a whole. It checks how the components work and interact with each other. Integration testing tries to find error in the communication and interaction between modules. For this system, the encoded and decoded modules were tested as whole system using various test cases.