Stegano
Design Document

By

Sean Reidy

C00227196

Institute of Technology Carlow

April 2021

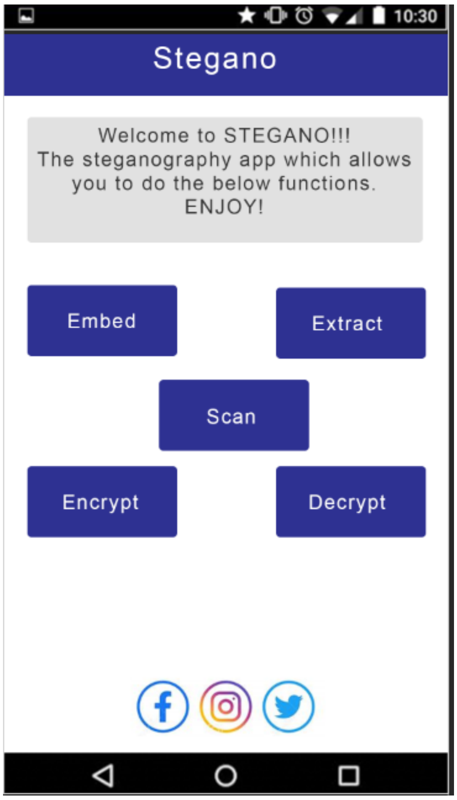
Contents

Introduction

This design document will cover the design functions and features of the stegano application. This document will provide detailed use cases and sequence diagrams. Included below are sample gui designs for each main function of the app. These will closely mimic the design in the final product. The gui for each function was made by using adobe photoshop. Since we are using Android Studio to create the app, we have to keep in mind that this is a standard android app, the gui interface has to follow this too. Android studio has limit capabilities when it comes down to designing the app. Below there are also sequence diagrams to help give the reader a deeper understanding into each function and how the app will be built. These diagrams should highlight the flow between the user and the app. The stegano app is a standalone app and its main functionality was highlighted in the functional specification document. With the limited capabilities of Android Studio and the developer's abilities in mind, the user interface will look like the below images:

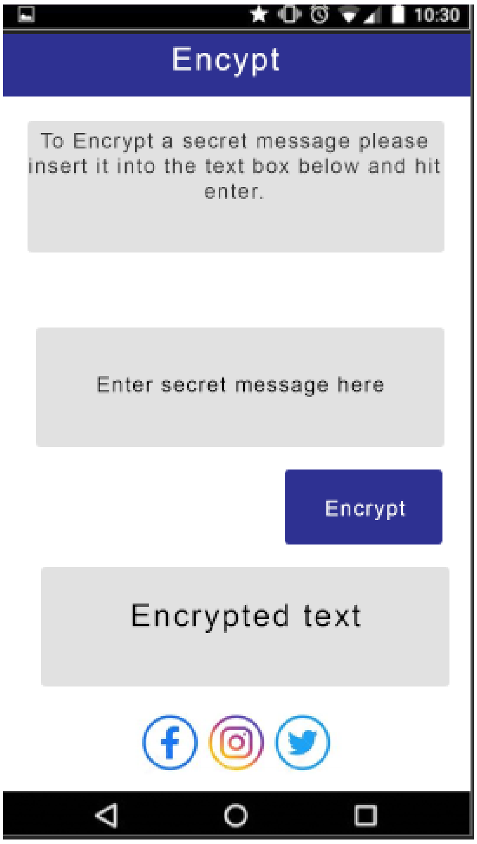
GUI design

Home Page



Welcome to STEGANO!!!
The steganography app which allows you to do the below functions.
ENJOY!

Encrypt Page



10:30

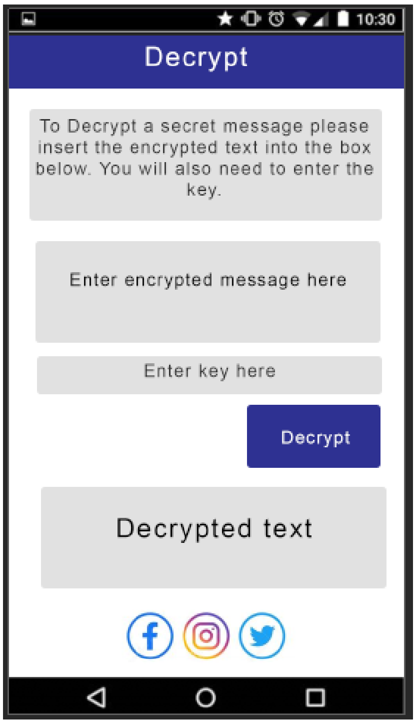
Encypt

To Encrypt a secret message please insert it into the text box below and hit enter.

Enter secret message here

Encrypted text

Decrypt Page



10:30

Decrypt

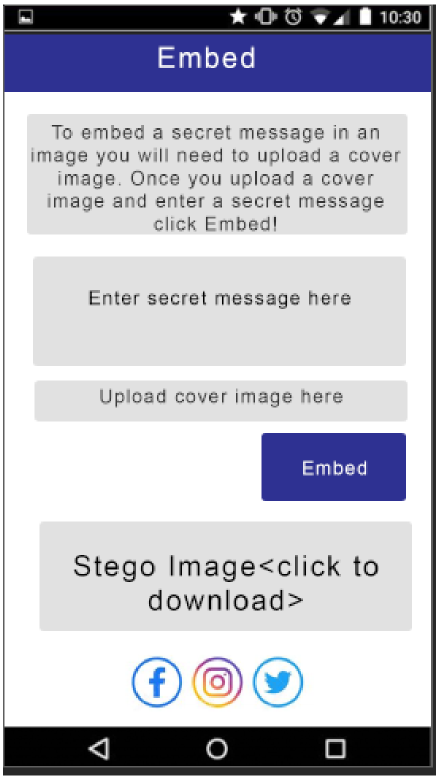
To Decrypt a secret message please insert the encrypted text into the box below. You will also need to enter the key.

Enter encrypted message here

Enter key here

Decrypted text

Embed Page



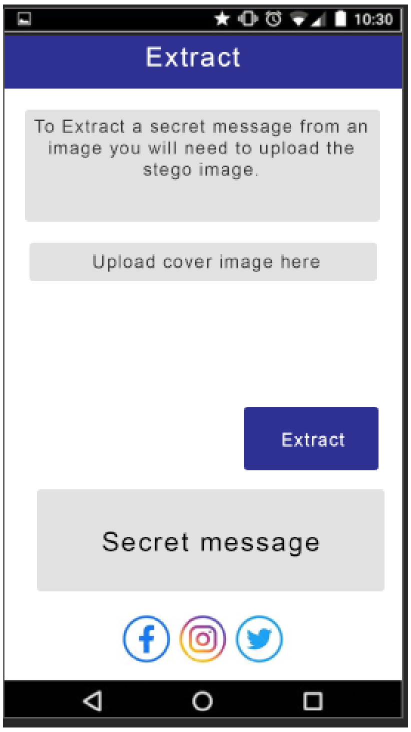
To embed a secret message in an image you will need to upload a cover image. Once you upload a cover image and enter a secret message click Embed!

Enter secret message here

Upload cover image here

Stego Image<click to download>

Extract Page

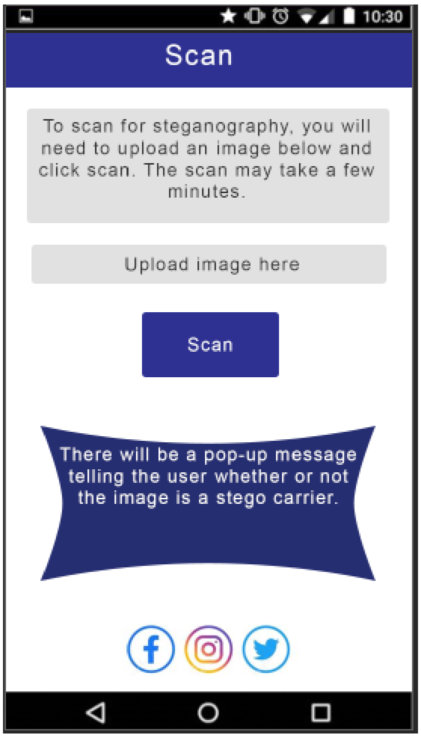


To Extract a secret message from an image you will need to upload the stego image.

Upload cover image here

Secret message

Scan Page



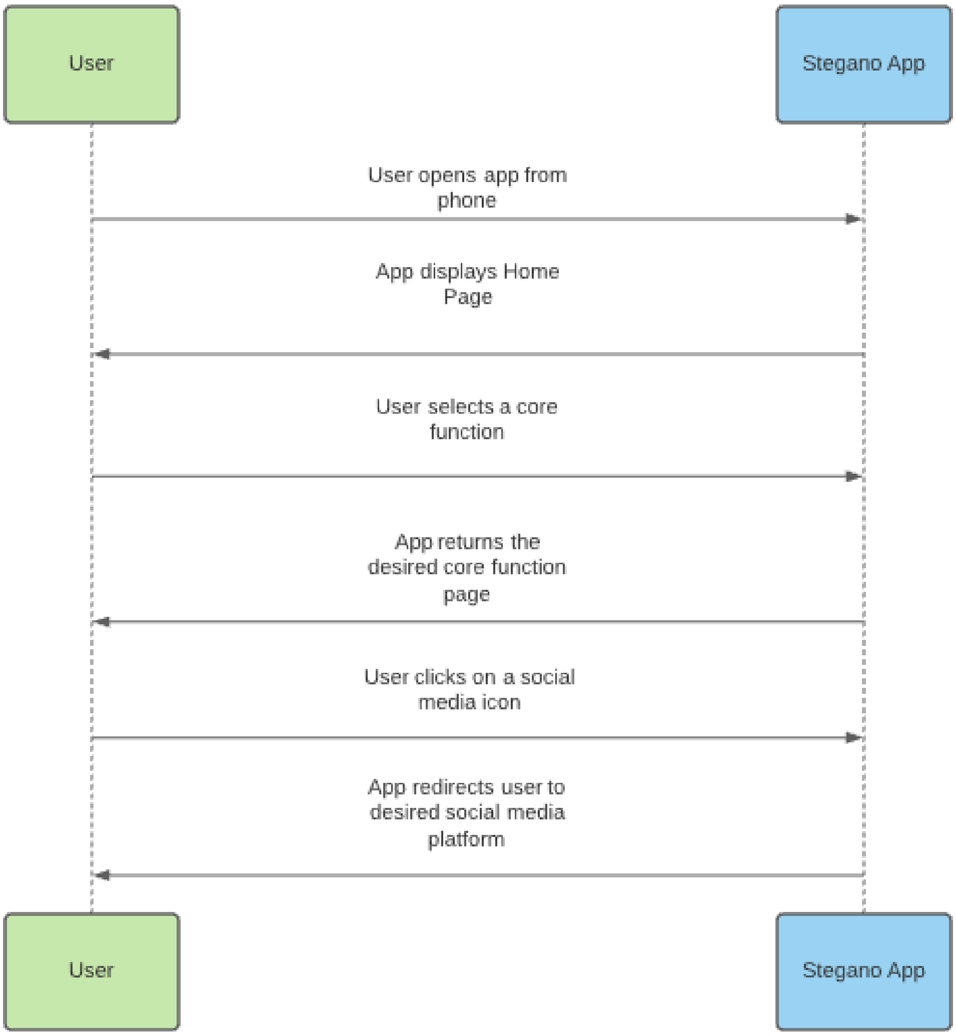
To scan for steganography, you will need to upload an image below and click scan. The scan may take a few minutes.

There will be a pop-up message telling the user whether or not the image is a stego carrier.

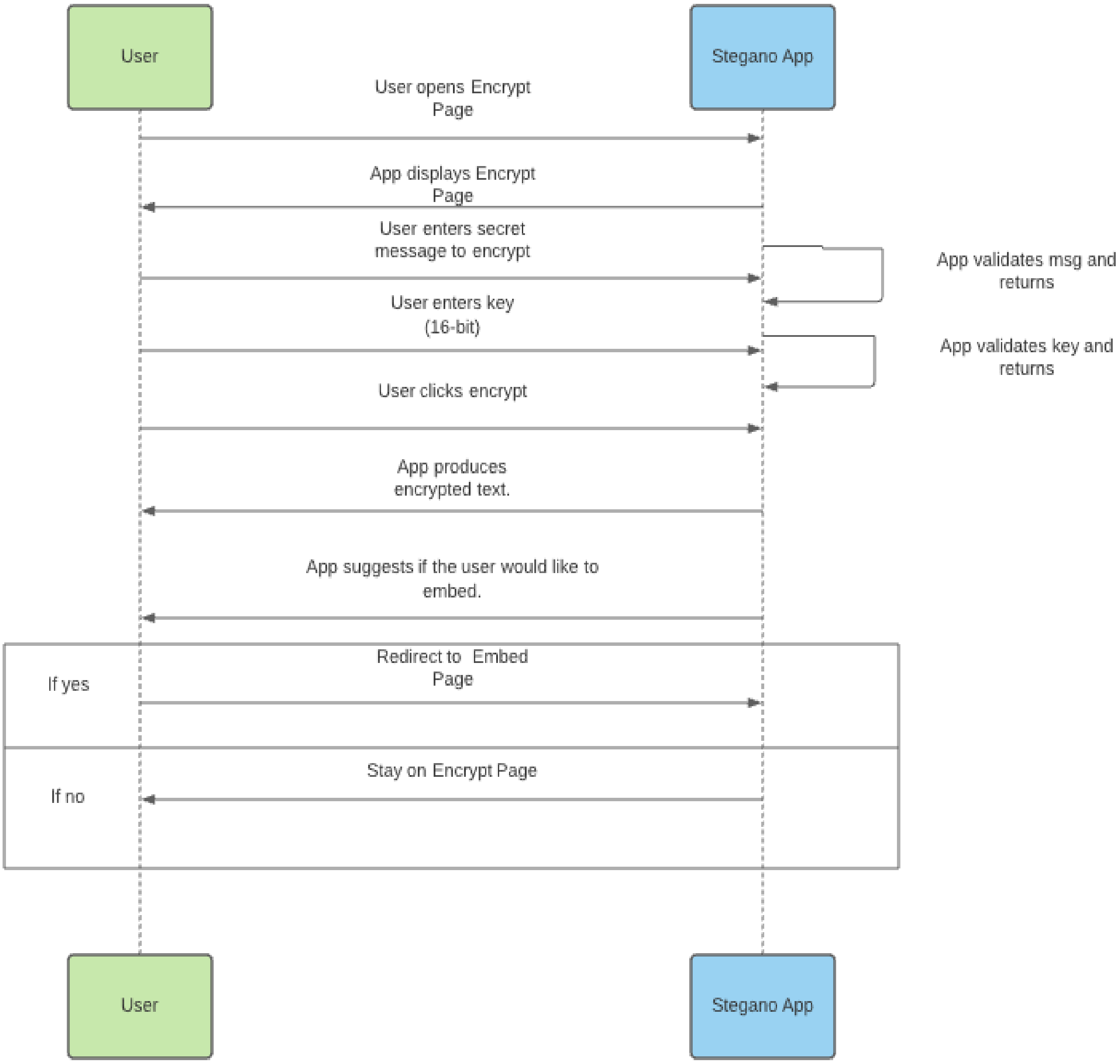
Sequence Diagrams

The sequence diagrams are here to highlight the flow between the app and the user for each core function mentioned above.

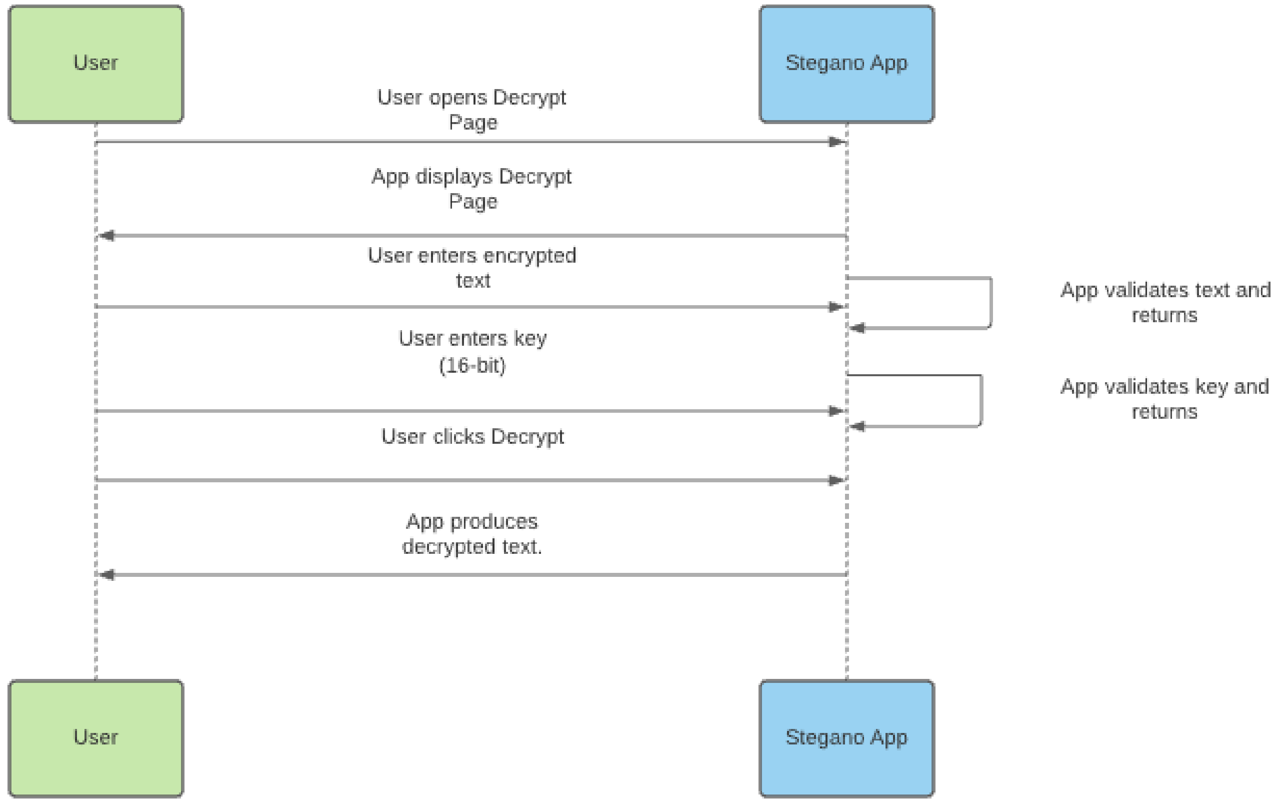
Home Page


User
Stegano App
User opens app from phone
App displays Home Page
User selects a core function
App returns the desired core function page
User clicks on a social media icon
App redirects user to desired social media platform
User
Stegano App

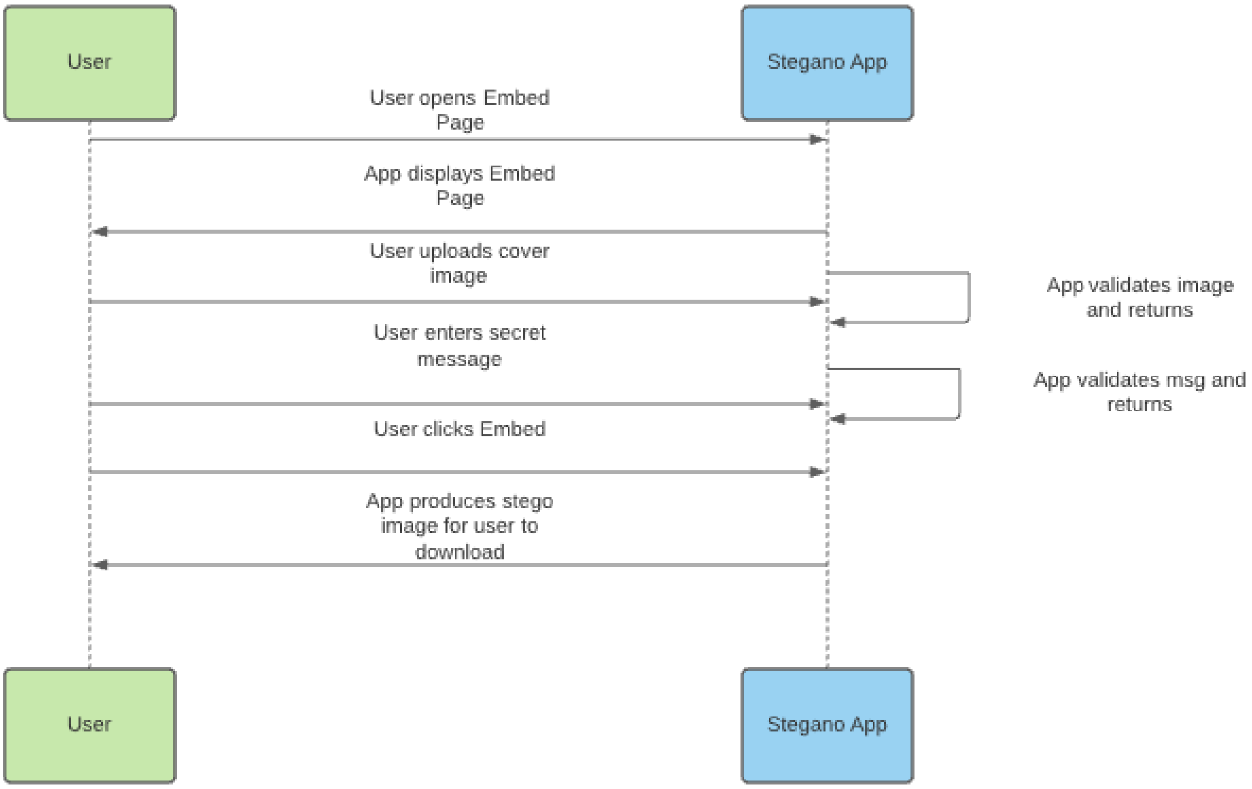
Encrypt Page


User
Stegano App
User opens Encrypt Page
App displays Encrypt Page
User enters secret message to encrypt
App validates msg and returns
User enters key (16-bit)
App validates key and returns
User clicks encrypt
App produces encrypted text.
App suggests if the user would like to embed.
If yes
Redirect to Embed Page
If no
Stay on Encrypt Page
User
Stegano App

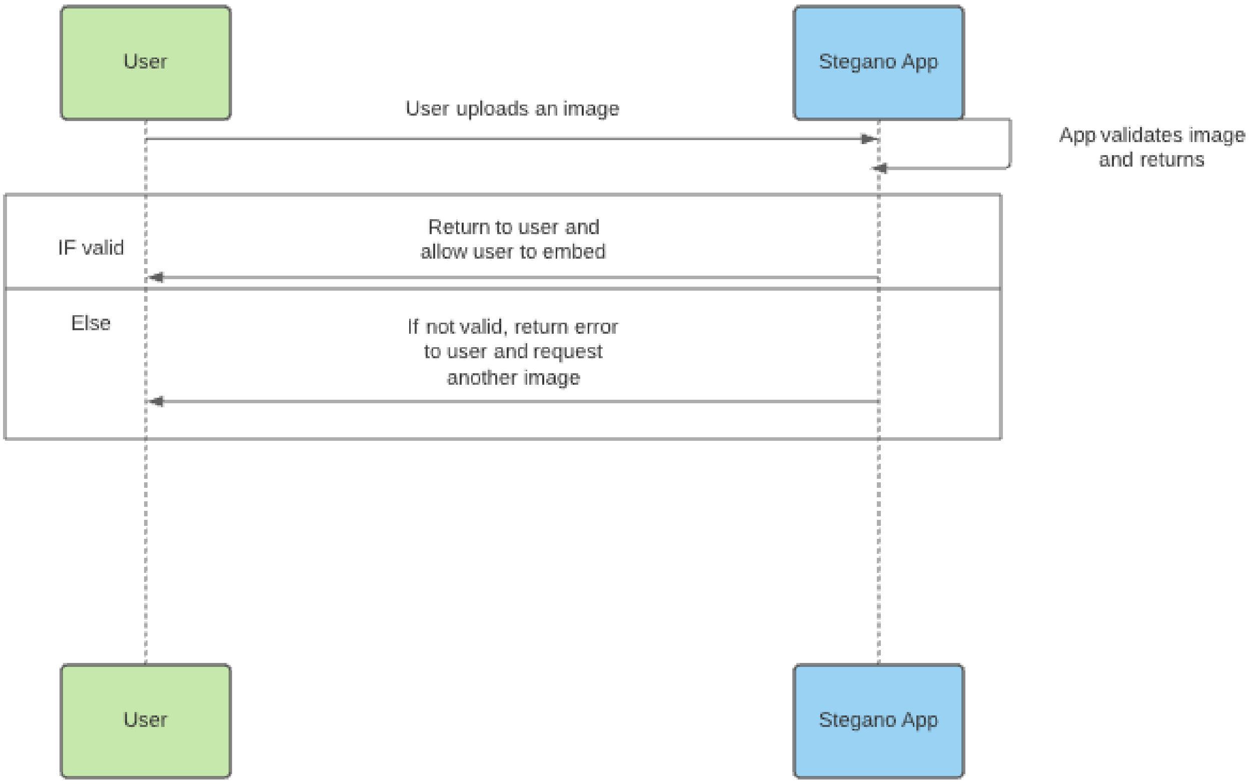
Decrypt Page


User
Stegano App
User opens Decrypt Page
App displays Decrypt Page
User enters encrypted text
App validates text and returns
User enters key (16-bit)
App validates key and returns
User clicks Decrypt
App produces decrypted text.
User
Stegano App

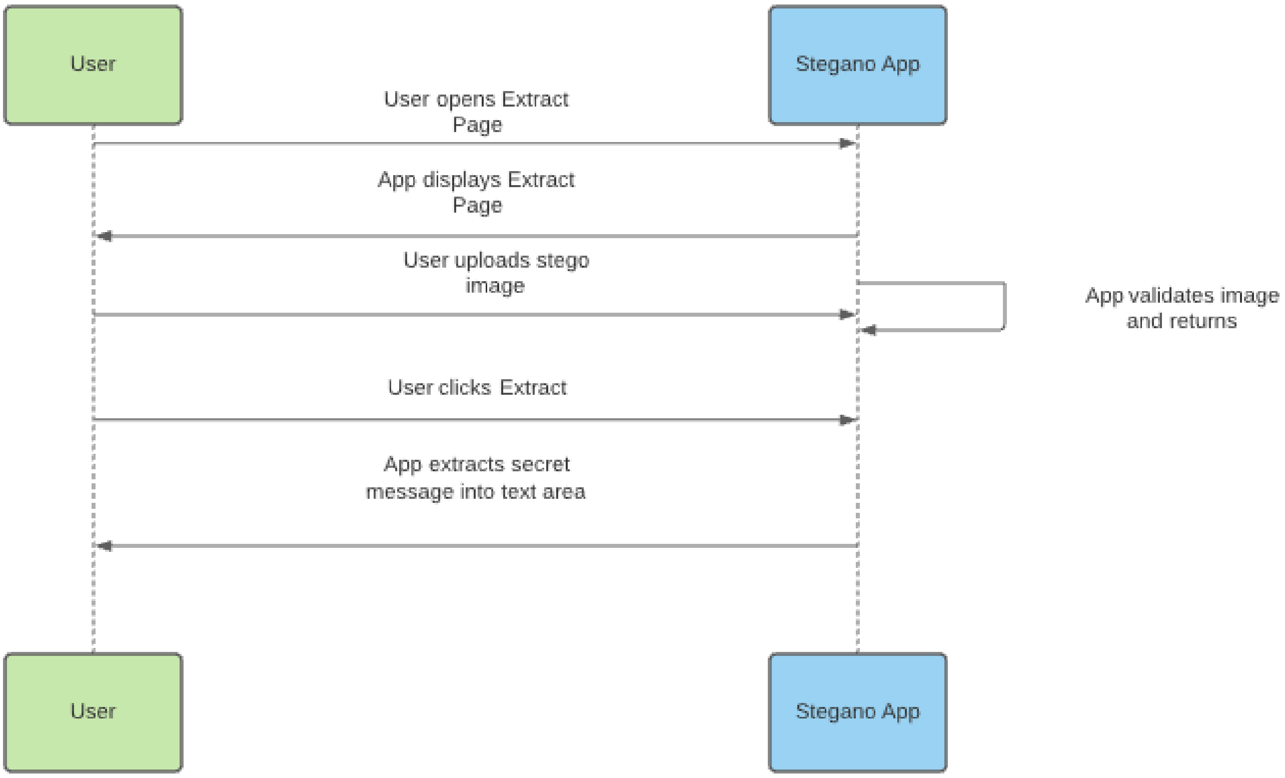
Embed Page


User
Stegano App
User opens Embed Page
App displays Embed Page
User uploads cover image
App validates image and returns
User enters secret message
App validates msg and returns
User clicks Embed
App produces stego image for user to download
User
Stegano App

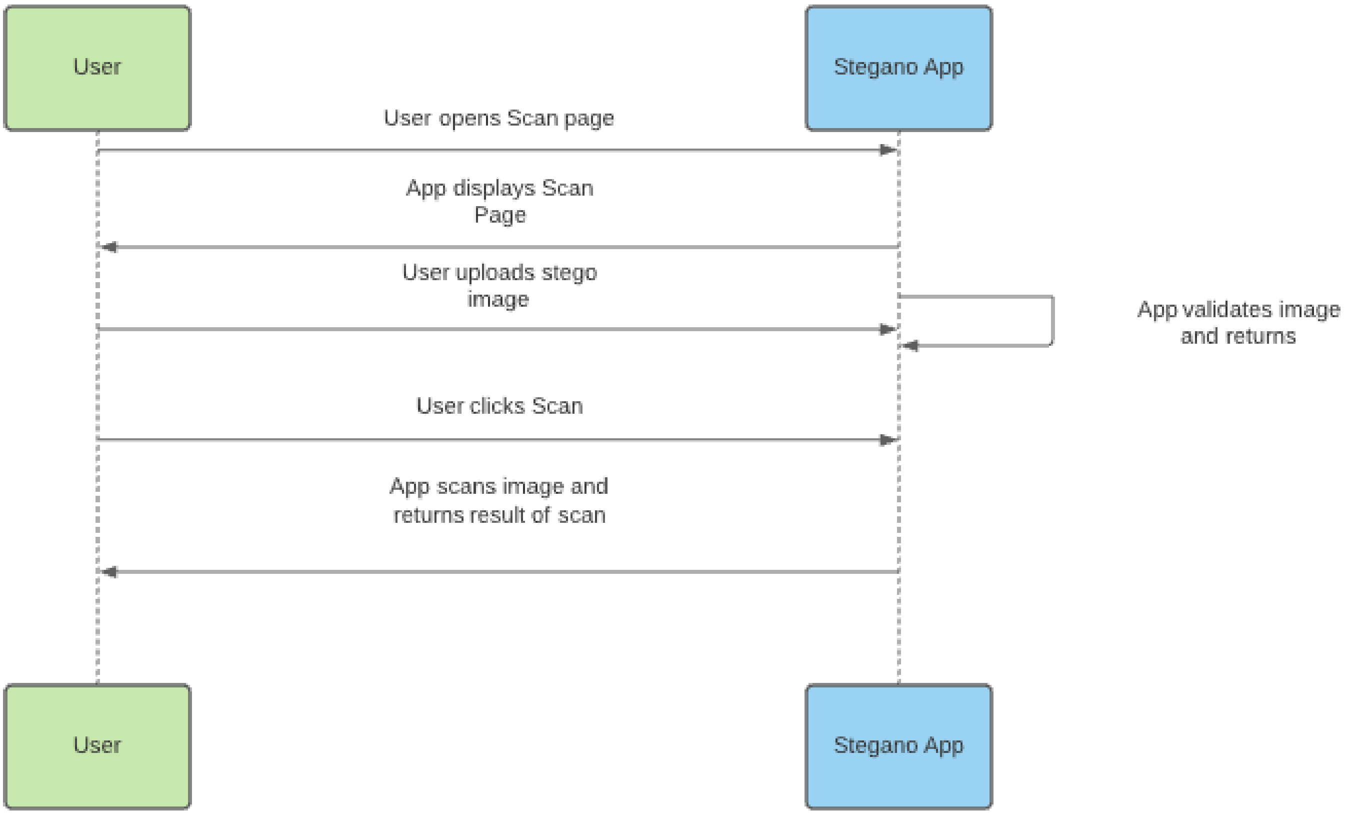
Upload image check1.2


User
Stegano App
User uploads an image
App validates image and returns
IF valid
Return to user and allow user to embed
Else
If not valid, return error to user and request another image
User
Stegano App

Extract Image


User
Stegano App
User opens Extract Page
App displays Extract Page
User uploads stego image
App validates image and returns
User clicks Extract
App extracts secret message into text area
User
Stegano App

Scan Page



Pseudo code

Embed

The embed function is used to embed a secret message into the LSB of an uploaded image. First step is to convert the secret message data from a decimal format to binary format. Then convert the cover image from decimal to binary too.

Input: cover image, secret message, secret key ///// secret is not required unless the string is encrypted.

Convert secret message into binary

Convert secret key into binary

Break down each byte of secret message and key into bits.

Replace last bit of cover image with a new bit.

Repeat process until no bits are left to insert.

Output: stego image

Extract

The extract function is used to extract a secret message from a stego image.

Input: stego image, secret key //////// secret is not required unless the secret message was encrypted.

Break down byte of stego image into bits

Extract the last bit and store it.

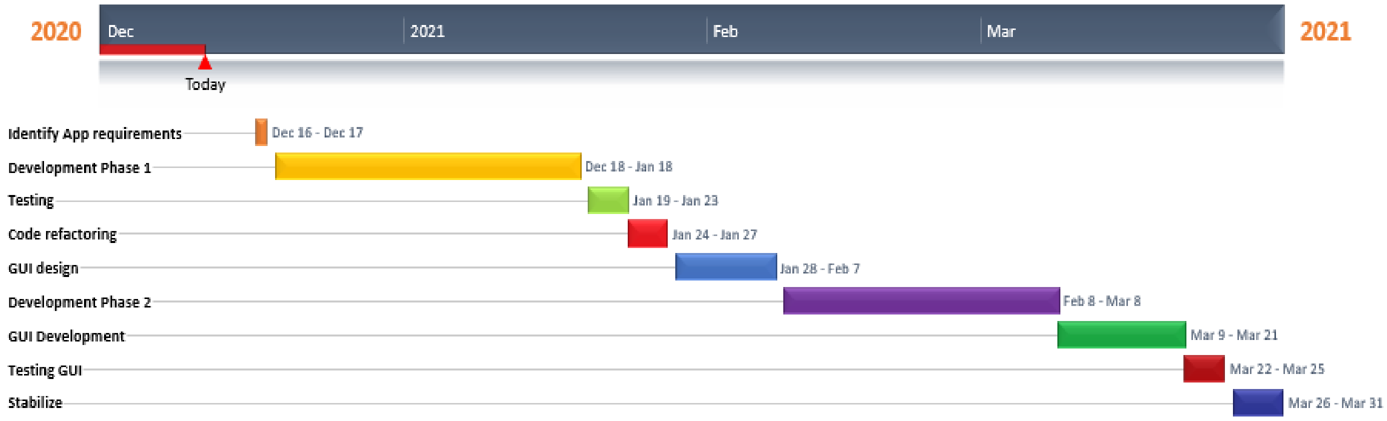
Repeat process for the remaining bits.

Convert the gathered bits from binary to decimal.

Output: secret text

Project Plan

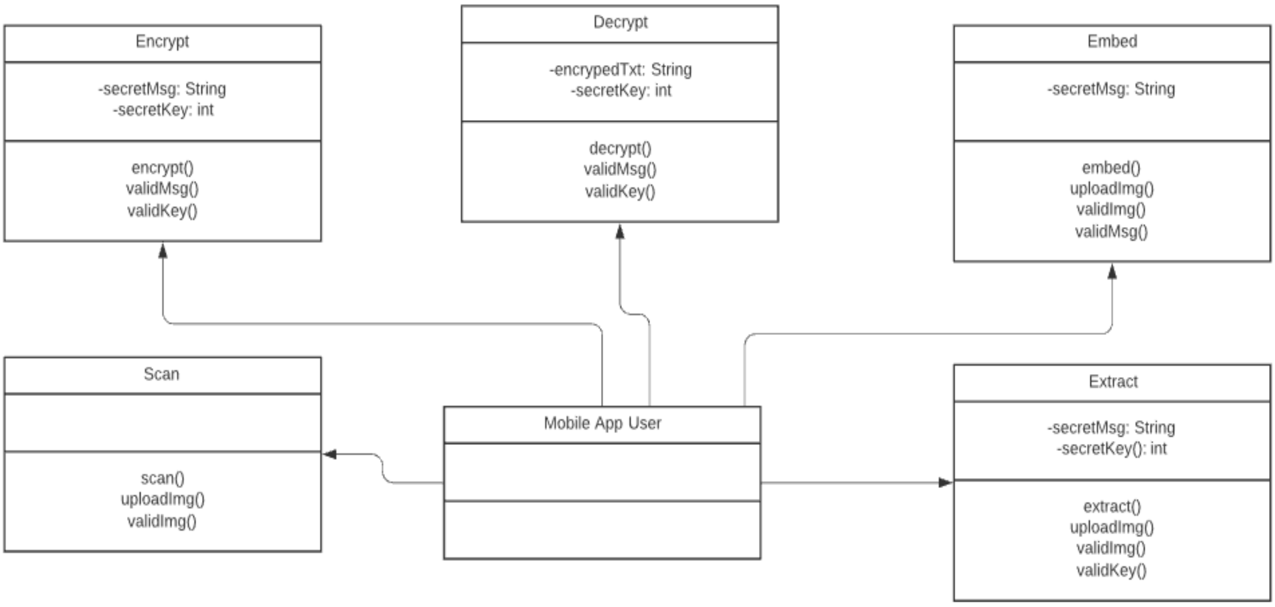
The project plan is also known as a project management plan. Below is a gantt chart made using office timeline. The gantt chart shows us the dedicated timeline for the development cycle of this project. Below we can see the deliverables and the corresponding due dates.



Class Diagram

In software development, a class diagram is a type of static structure diagram that describes the structure of the system. It does this by showing the classes, their attributes, functions and relationships among different objects.

With regards to this project and since the stegano app is a standalone app, a class diagram is suited well to describe the relationship between the user and the app. The class diagram is below:


Encrypt
-secretMsg: String
-secretKey: int
encrypt()
validMsg()
validKey()
Decrypt
-encrypedTxt: String
-secretKey: int
decrypt()
validMsg()
validKey()
Embed
-secretMsg: String
embed()
uploadImg()
validImg()
validMsg()
Scan
scan()
uploadImg()
validImg()
Mobile App User
Extract
-secretMsg: String
-secretKey(): int
extract()
uploadImg()
validImg()
validKey()