

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN - ĐẠI HỌC QUỐC GIA TP HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**NHẬP MÔN MÃ HOÁ MẬT MÃ - 19MMT**

## **ĐỒ ÁN 1**

## **HỆ THỐNG LƯU TRỮ FILE**



**GIÁO VIÊN BỘ MÔN**

NGUYỄN ĐÌNH THỨC

NGÔ ĐÌNH HY

NGUYỄN VĂN QUANG HUY

### **NHÓM STACKOVERNIGHT**

BẠCH MINH KHÔI	19127181
PHẠM DUY TIẾN	19127577
PHẠM NGUYỄN TƯỜNG VY	19127636
TRƯƠNG BỬU Ý	19127638

# MỤC LỤC

<b>MÔI TRƯỜNG VÀ CÀI ĐẶT</b>	<b>4</b>
Thư viện Python	4
Hệ thống lưu trữ	4
HEROKU CLEARDB MYSQL	4
FIREBASE	6
Deploy server bằng Heroku	8
<b>SƠ ĐỒ HOẠT ĐỘNG</b>	<b>9</b>
Đăng nhập	9
Đăng ký	9
Upload 1 ảnh	10
Tải 1 ảnh	10
Tải tất cả ảnh	11
Lấy public key	11
Chia sẻ ảnh	12
<b>THUẬT TOÁN - MÃ GIẢ</b>	<b>13</b>
Mã hoá	13
Thuật toán	13
Mã giả	13
Giải mã	13
Thuật toán	13
Mã giả	14
<b>HƯỚNG DẪN SỬ DỤNG CHƯƠNG TRÌNH</b>	<b>14</b>
Biên dịch và chạy mã nguồn	14
Tạo tài khoản	15
Đăng nhập	16
Upload ảnh lên server	18
Mã hóa	19
Chia sẻ ảnh	19
Tải ảnh	22
Giải mã	22
<b>TỰ ĐÁNH GIÁ</b>	<b>24</b>

Ưu điểm	24
Nhược điểm	24
Mở rộng	24
<b>TÀI LIỆU THAM KHẢO</b>	<b>25</b>

# 1. MÔI TRƯỜNG VÀ CÀI ĐẶT

Toàn bộ hệ thống (FE và BE) được cài đặt trên windows 10 ngôn ngữ python, kết hợp với các hệ thống lưu trữ (heroku, mysql).

## 1.1. Thư viện Python

Backend:

- Flask
- Firebase
- Query
- Flask\_restful
- Json
- Zipfile
- Os
- Bcrypt
- Mysql.connector
- Tempfile
- Logging

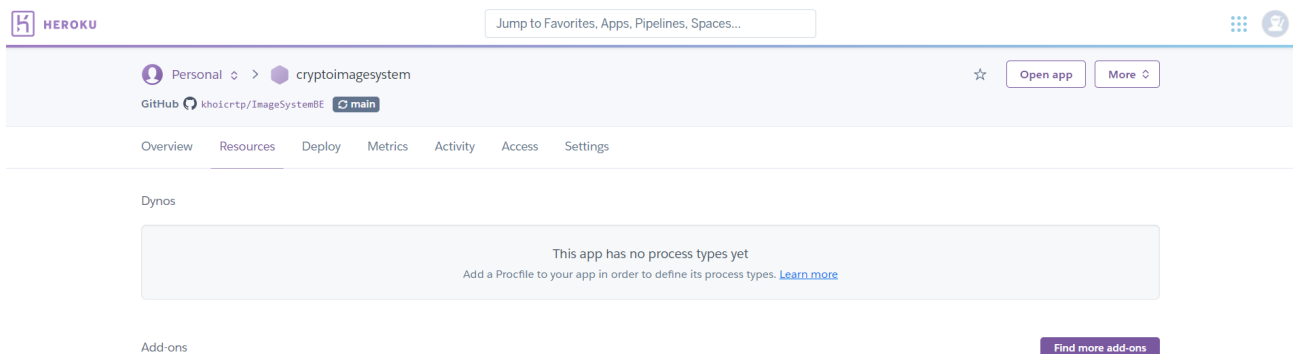
Frontend: (FE check lại lib xem có dư hay thiếu gì không nha)

- Tkinter
- Platform
- Requests
- Json
- Numpy
- PIL
- Time
- Os
- matplotlib
- app

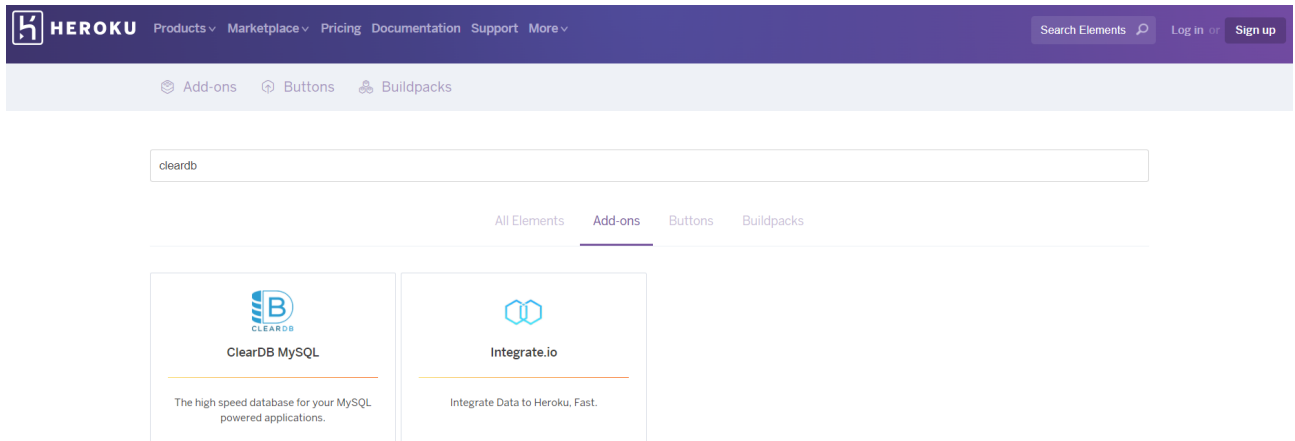
## 1.2. Hệ thống lưu trữ

### 1.2.1. HEROKU CLEARDB MYSQL

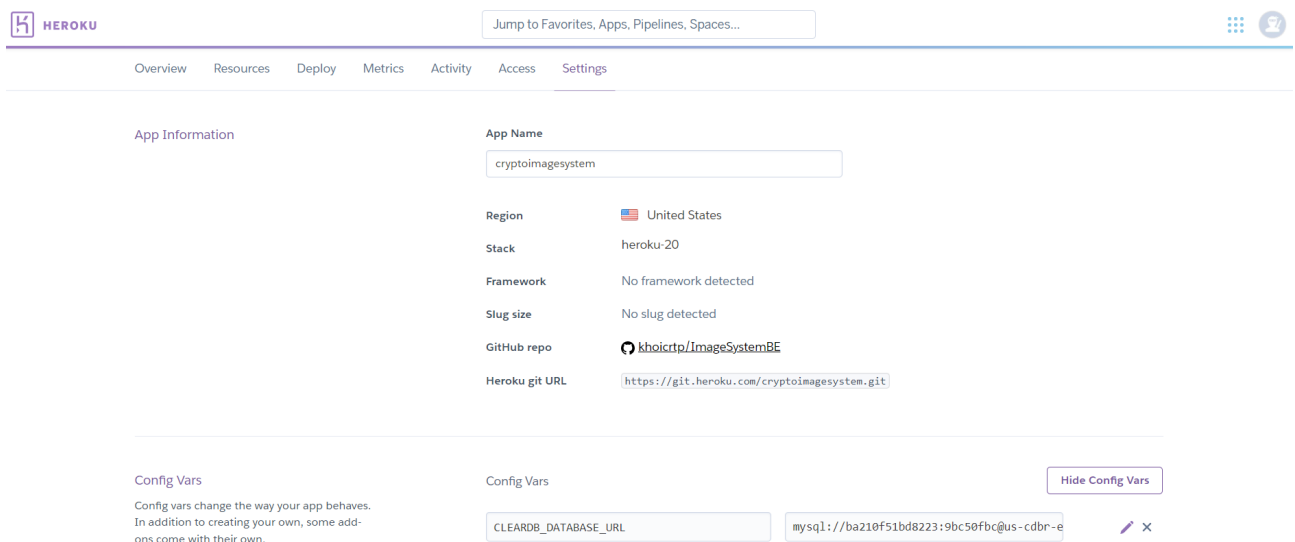
- Tạo tài khoản heroku, tạo empty project.
- Vào mục Resources và nhấn nút **Find more add-ons**



- Tìm add-ons ClearDB MySQL và thêm vào project



- Sau khi thêm add-ons vào project, vào tab Setting để lấy Config Vars cho database



- Config vars của database có dạng:  
**mysql://ba210f51bd8223:9bc50fbc@us-cdbr-east-05.cleardb.net/heroku\_d08923bbc460fa4?reconnect=true**

Trong đó:

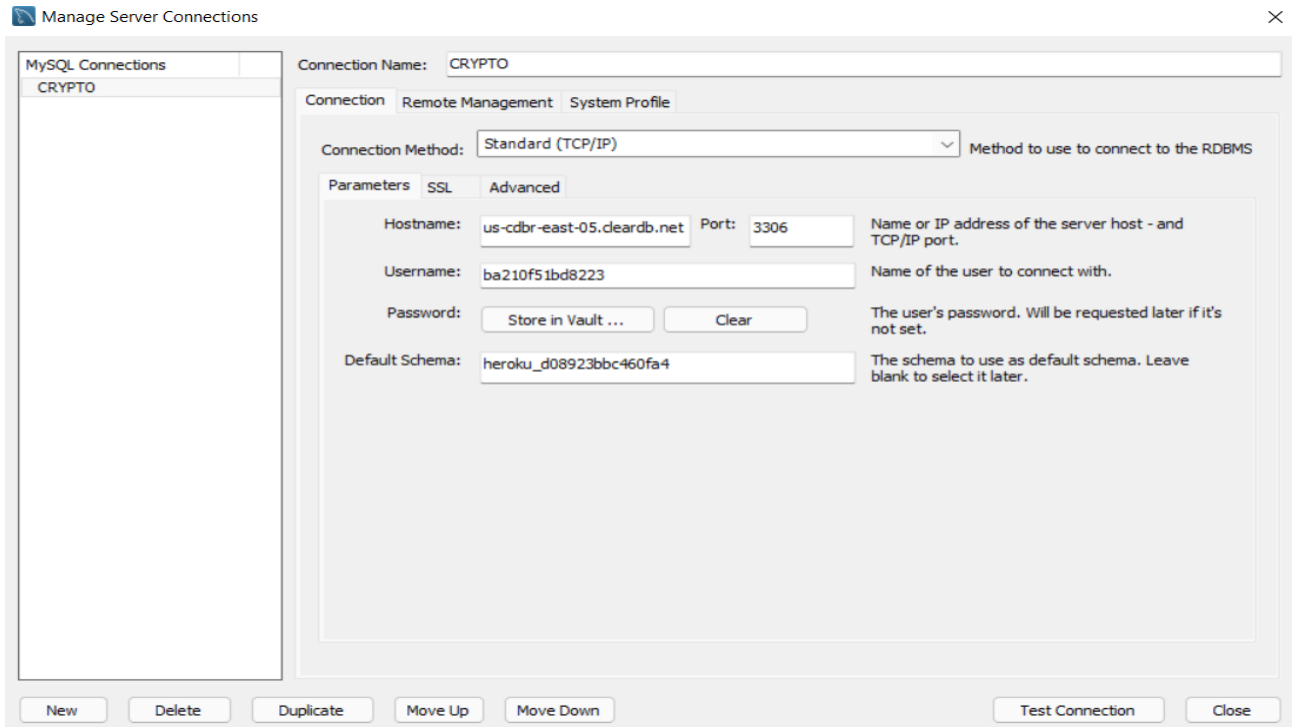
**Username:** ba210f51bd8223

**Password:** 9bc50fbc

**Hostname:** us-cdbr-east-05.cleardb.net

**Default Schema:** heroku\_d08923bbc460fa4

- Sử dụng phần mềm **MySQL Workbench** quản lý database:

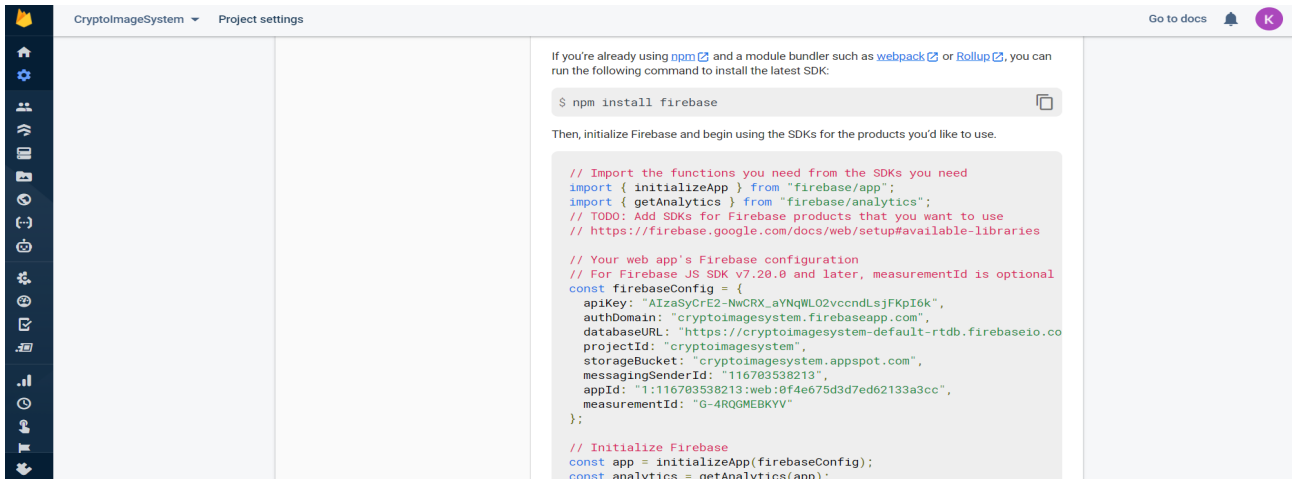


## 1.2.2. FIREBASE

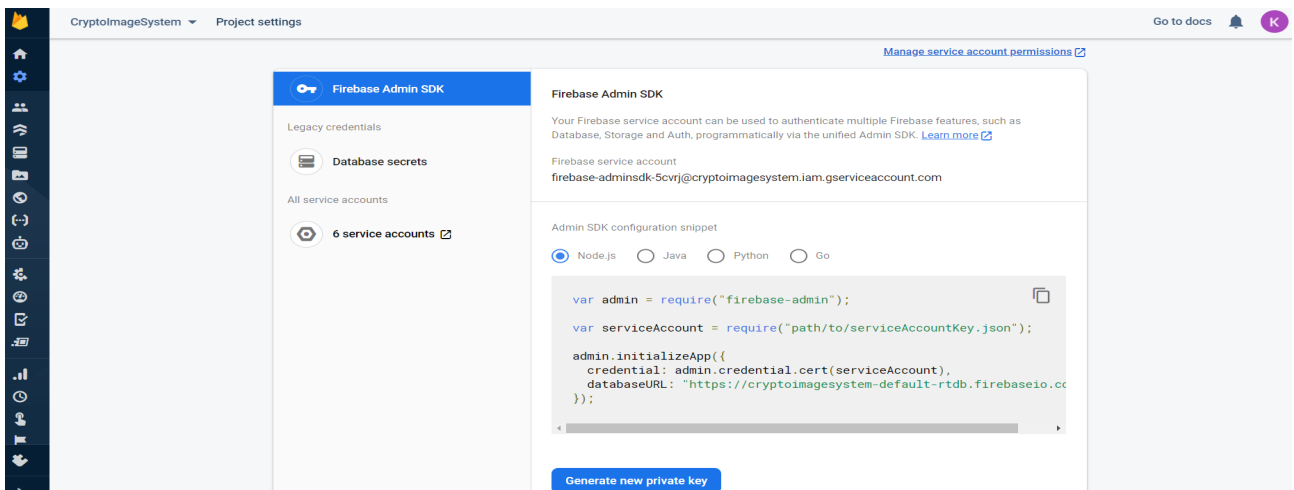
- Vào [Firebase console \(google.com\)](https://firebase.google.com/) để tạo project mới.
- Vào tab Firebase Storage và thiết lập các quy định phù hợp để dễ dàng sử dụng



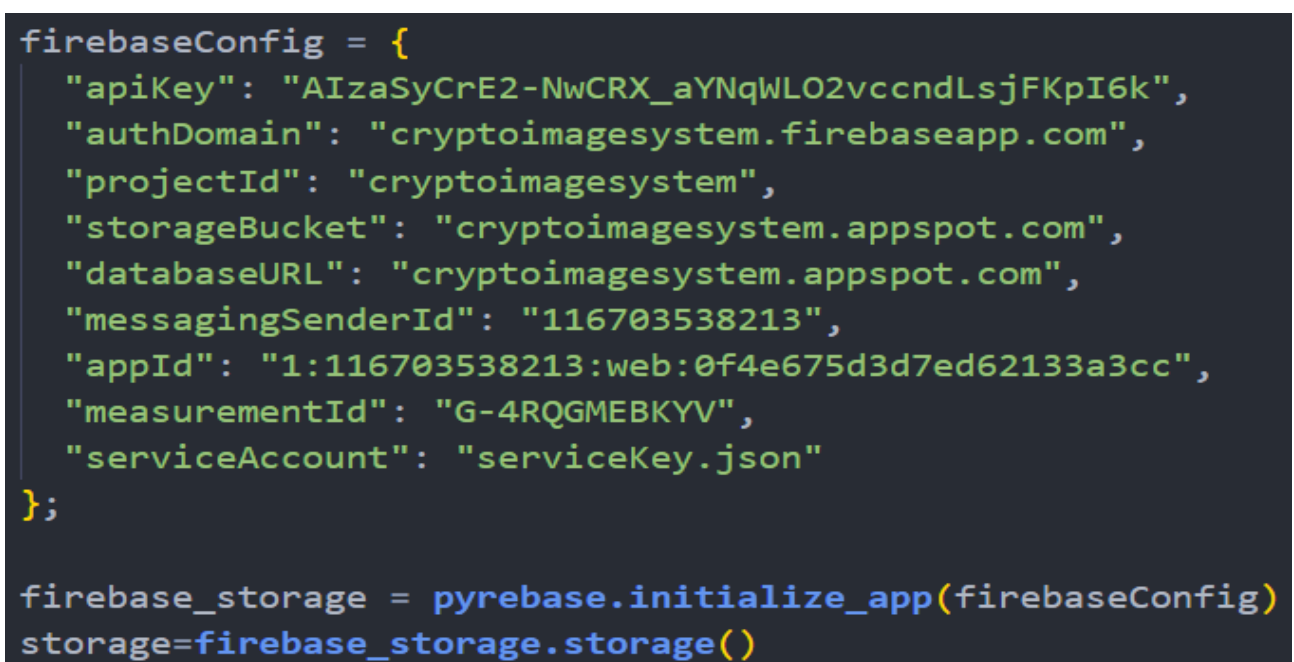
- Vào tab Project settings để lấy thông tin thiết lập của Firebase



- Tiếp đến, vào Firebase Admin SDK để lấy service key bằng nút **Generate new private key**



- Sử dụng thư viện pyrebase để thiết lập kết nối tới Firebase storage:



### 1.3. Deploy server bằng Heroku

- Tải và cài đặt Heroku CLI [The Heroku CLI | Heroku Dev Center](#)
- Tạo Procfile để Heroku có thể build app từ file app.py.

```
Procfile
You, 2 days ago | 1 author (You)
1 web: gunicorn app:app --preload --log-file -
```

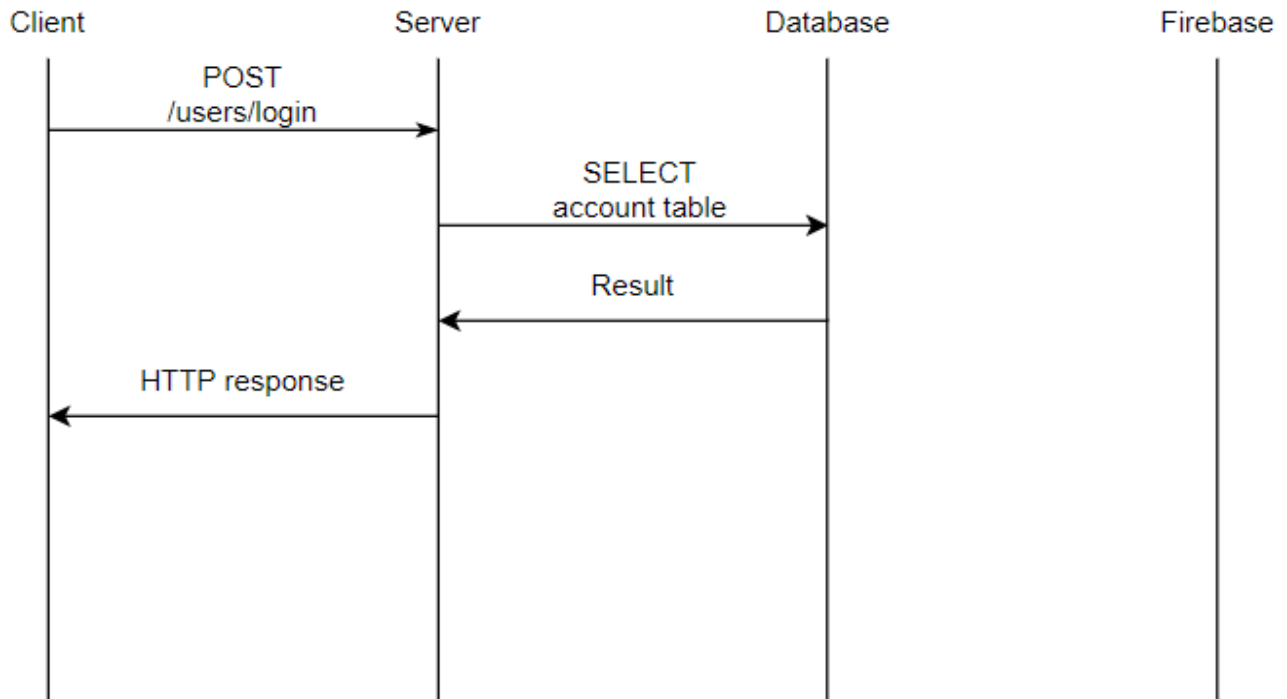
- Sử dụng Terminal lần lượt thực hiện:  
\$ heroku login: đăng nhập vào Heroku  
\$ heroku git:clone -a test-eflask-crypto: Clone Heroku repository.  
\$ cd test-eflask-crypto: Đổi directory tới thư mục vừa clone.  
\$ git add . : thêm tất cả file trong thư mục vào danh sách của git.  
\$ git commit -am "first commit": commit những thay đổi.  
\$ git push heroku master: push lên branch master của Heroku repository.  
- Khi thực hiện quá trình push lên Heroku, quá trình tự động build code và chạy bằng file Procfile sẽ được thực hiện tự động. Sau đó, Heroku sẽ cấp 1 URL đặt theo tên của Heroku repository được dùng để đưa vào sử dụng là [test-eflask-crypto.herokuapp.com](#).



## 2. SƠ ĐỒ HOẠT ĐỘNG

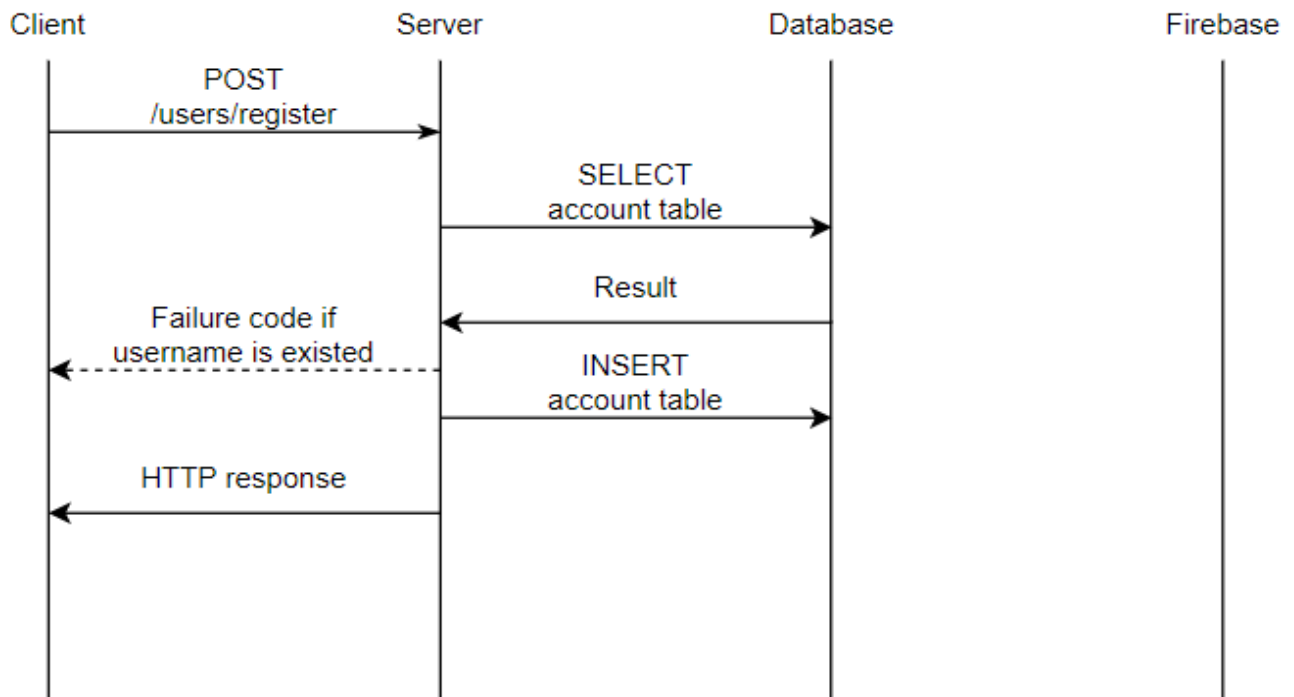
### 2.1. Đăng nhập

User gửi username, password tới server, server sẽ encode mã và so sánh với mã đã được mã hoá lưu trong database.



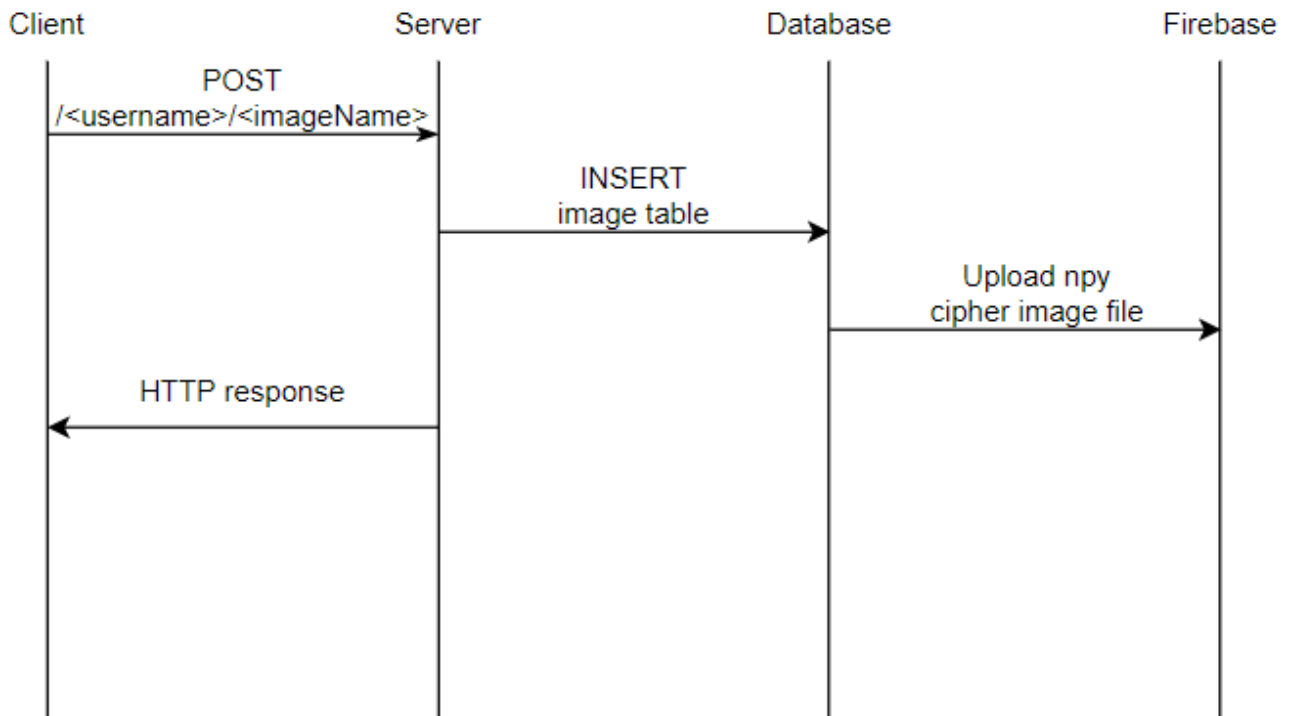
### 2.2. Đăng ký

User gửi username, password, publickey tới server, server sẽ kiểm tra username đã tồn tại hay chưa, sau đó mã hoá mật khẩu và lưu các thông tin vào database.



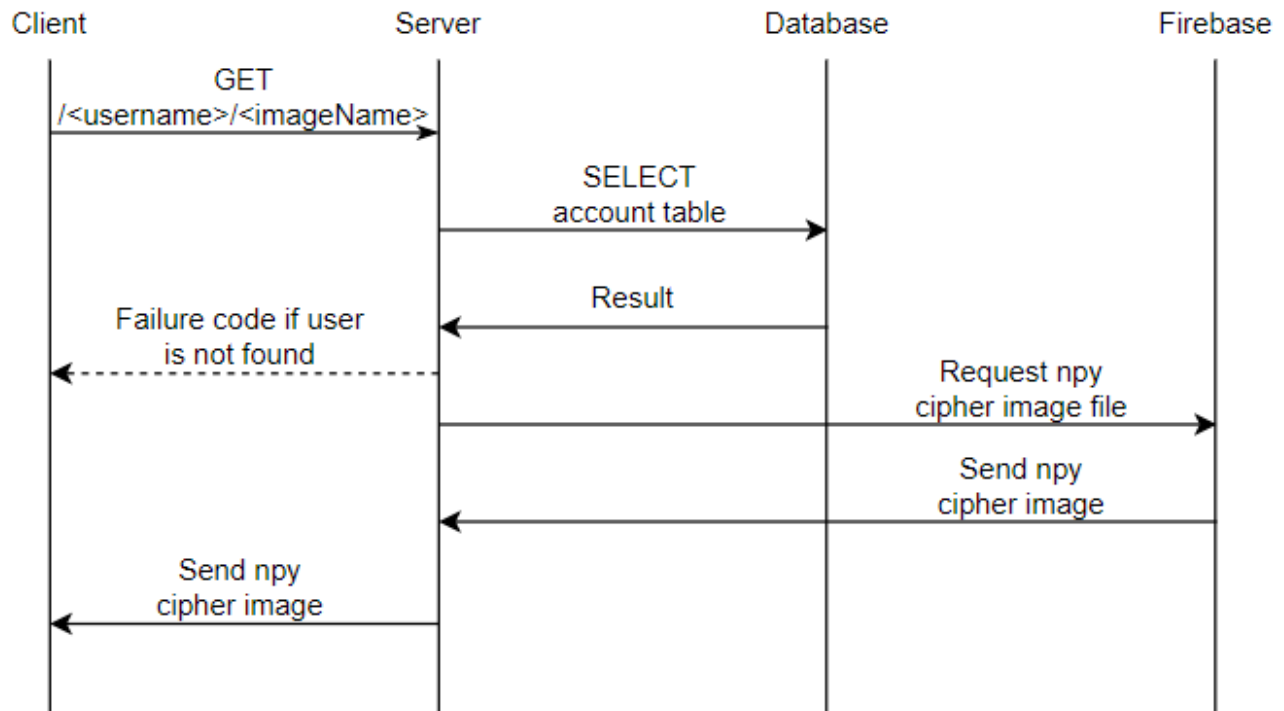
## 2.3. Upload 1 ảnh

Truy cập đến url và gửi kèm file ảnh, server sẽ lấy tên ảnh và username từ url, thực hiện việc lưu trữ ảnh (.npy) lên firebase và lưu tên ảnh cũng như tên file ảnh (.npy) vào database.



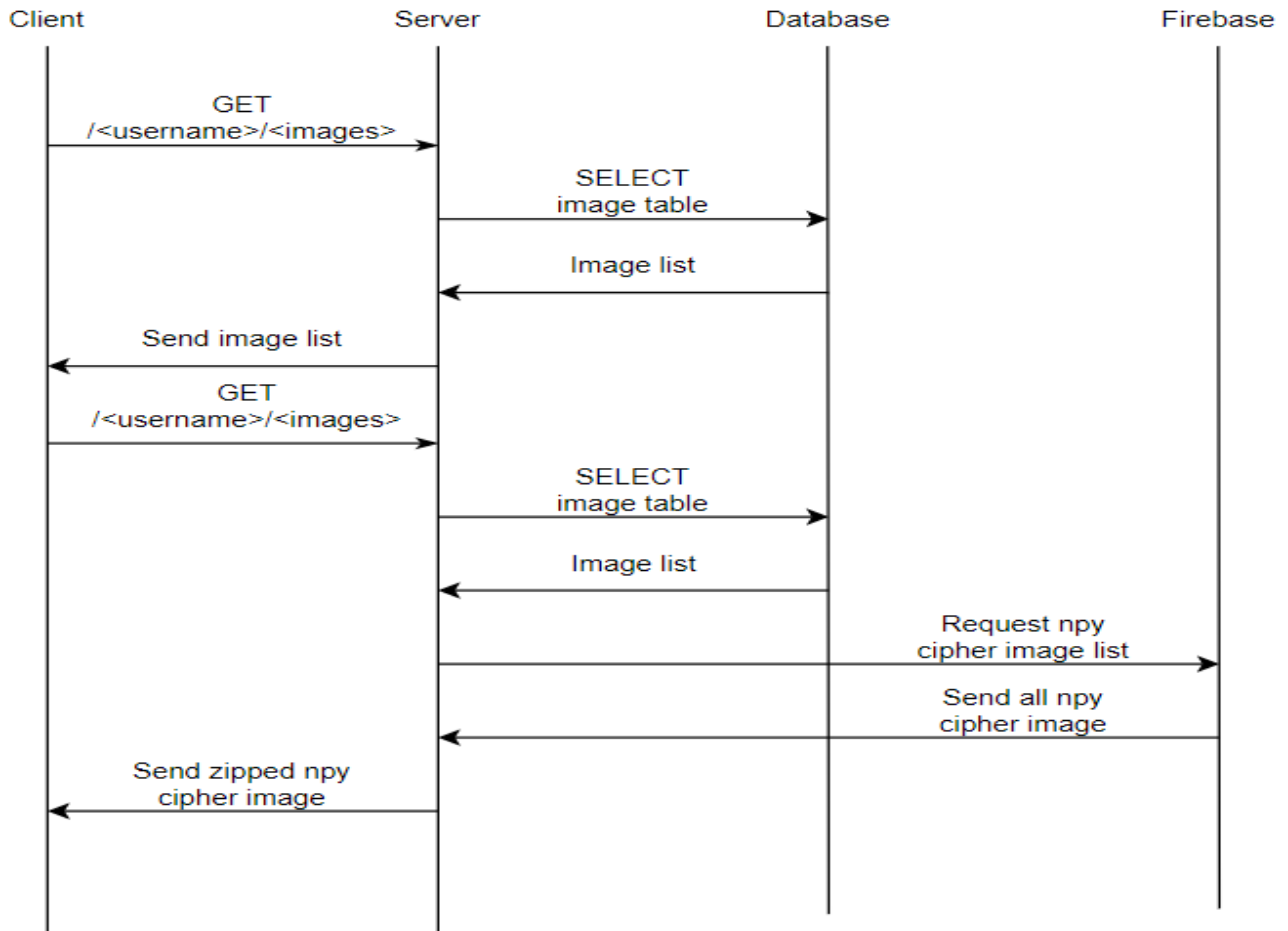
## 2.4. Tải 1 ảnh

Truy cập đến url, server sẽ lấy ảnh từ firebase về và gửi trả.



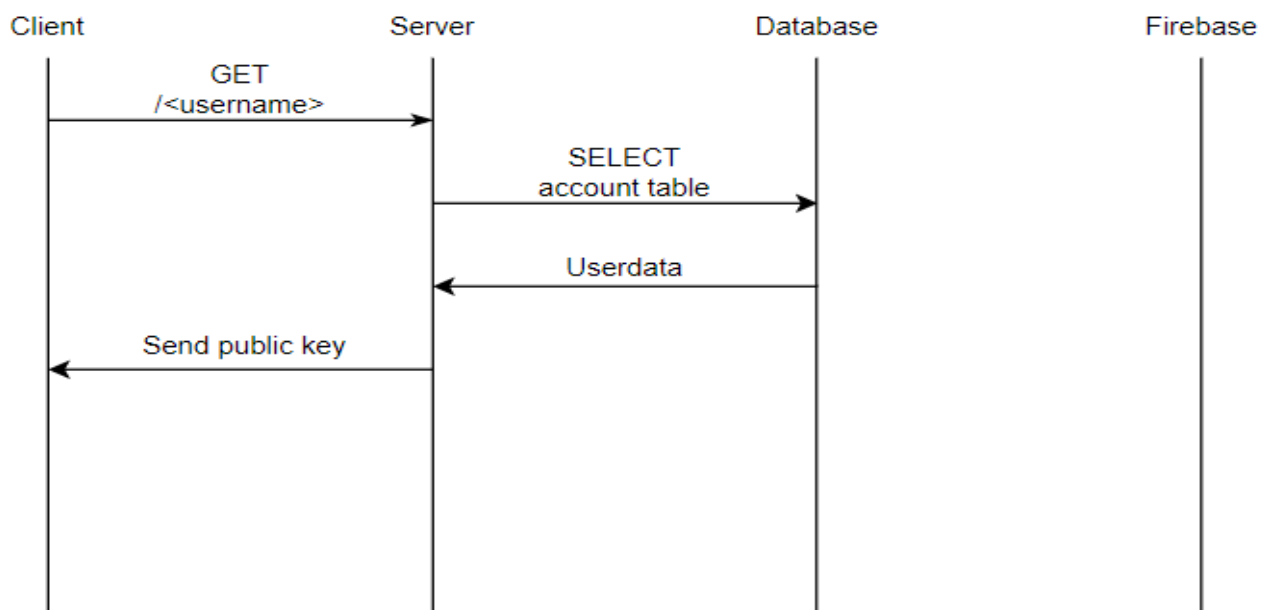
## 2.5. Tải tất cả ảnh

Truy cập đến url, server sẽ lấy ảnh từ firebase về và gửi trả.



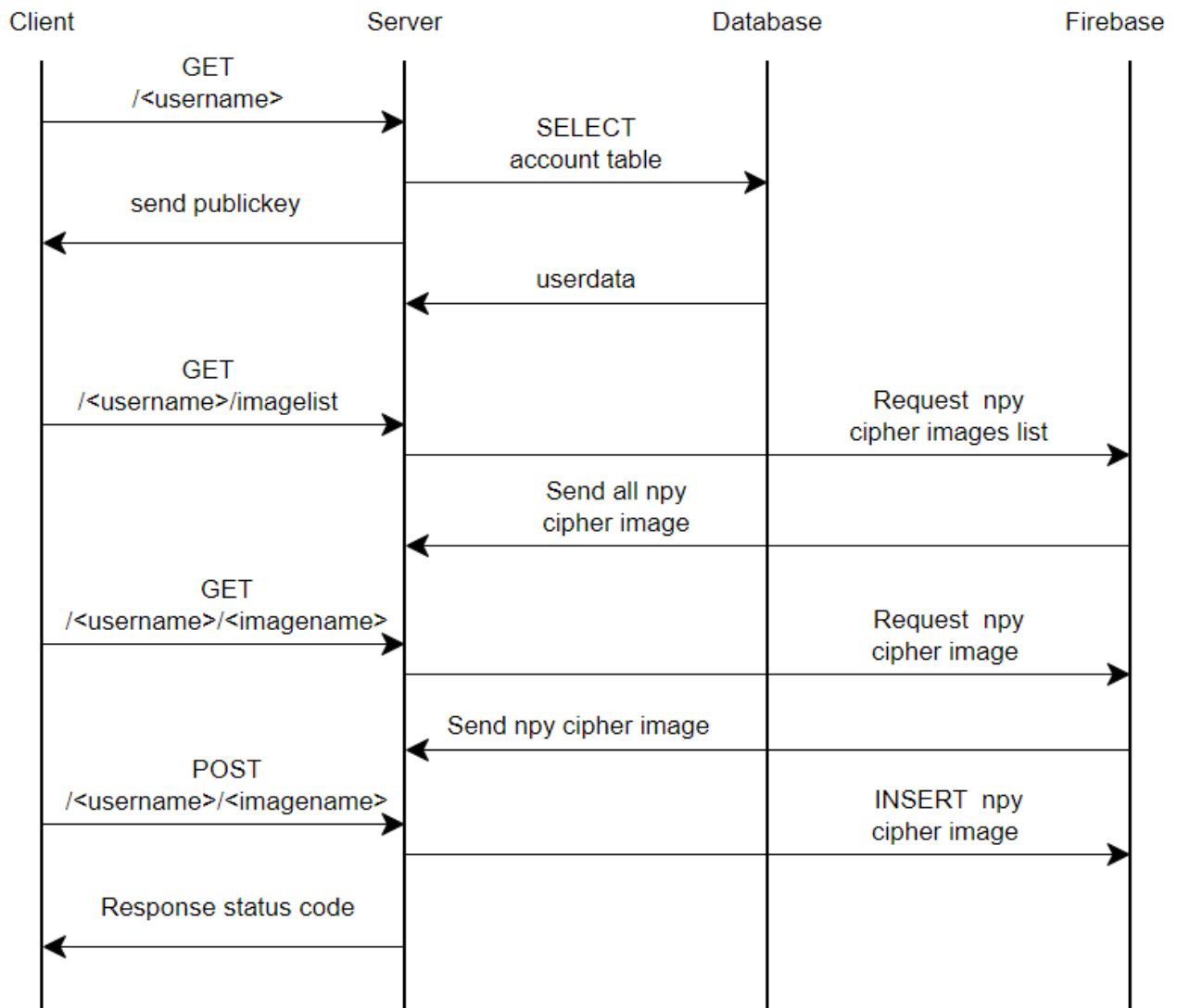
## 2.6. Lấy public key

Truy cập đến url chứa username cần lấy public key, server sẽ kiểm tra user có tồn tại hay không và trả về publickey tương ứng.



## 2.7. Chia sẻ ảnh

Gửi request đến server kèm username của người nhận để lấy public key của người nhận nhằm phục vụ việc mã hoá ảnh theo khoá của người nhận, giúp người nhận có thể tải bản rõ của ảnh được chia sẻ. Sau đó user sẽ lấy request lấy tên các tệp ảnh của mình. Tiếp đó, lần lượt gửi GET requests đến server để lấy nội dung ảnh mã hoá được lưu trên firebase về giải mã và mã hoá lại theo khoá của người nhận, sau đó gửi POST requests để upload ảnh được mã hoá vào folder của người nhận tương ứng trên firebase.



## 3. THUẬT TOÁN - MÃ GIẢ

### 3.1. Mã hoá

Thực hiện mã hoá trên từng điểm ảnh (RGB).

#### 3.1.1.1. Thuật toán

Input: mảng 3 chiều của ảnh, public key.

Bước 1: Khởi tạo 2 mảng 3 chiều theo size ảnh.

Bước 2: Mã hoá trên từng điểm ảnh (RGB) với publickey, với điểm ảnh raw (chưa module 256 - chưa chuẩn RGB) và điểm ảnh chuẩn RGB vào 2 mảng tương ứng.

Bước 3: Trả về 2 mảng 3 chiều tìm được từ bước 2.

#### 3.1.1.2. Mã giả

**def encrypt(image, e, n):**

    Khởi tạo encrypted\_image, encrypted.

    for i = 0, j = 0, to image.rows, image.columns:

        r, g, b = image[i][j]

        C\_R =  $r^e \bmod n$

        C\_G =  $g^e \bmod n$

        C\_B =  $b^e \bmod n$

        encrypted[i][j] = [C\_R, C\_G, C\_B]

        C\_R = C\_R % 256

        C\_G = C\_G % 256

        C\_B = C\_B % 256

        encrypted\_image[i,j] = [C\_R, C\_G, C\_B]

    return encrypted\_image, encrypted

### 3.2. Giải mã

Thực hiện giải mã trên từng điểm ảnh.

#### 3.2.1.1. Thuật toán

Input: mảng 3 chiều được mã hoá raw, private key.

Bước 1: Khởi tạo 1 mảng 3 chiều với size = input.

Bước 2: Giải mã trên từng điểm ảnh (RGB) với private key.

Bước 3: Trả về mảng 3 chiều tìm được từ bước 2.

### 3.2.1.2. Mã giả

**def decrypt(encrypted, d, n):**

    Khởi tạo mảng 3 chiều image.

    for i = 0, j = 0, to image.rows, image.columns:

        r, g, b = encrypted[i][j]

$M\_R = r^d \bmod n$

$M\_G = g^d \bmod n$

$M\_B = b^d \bmod n$

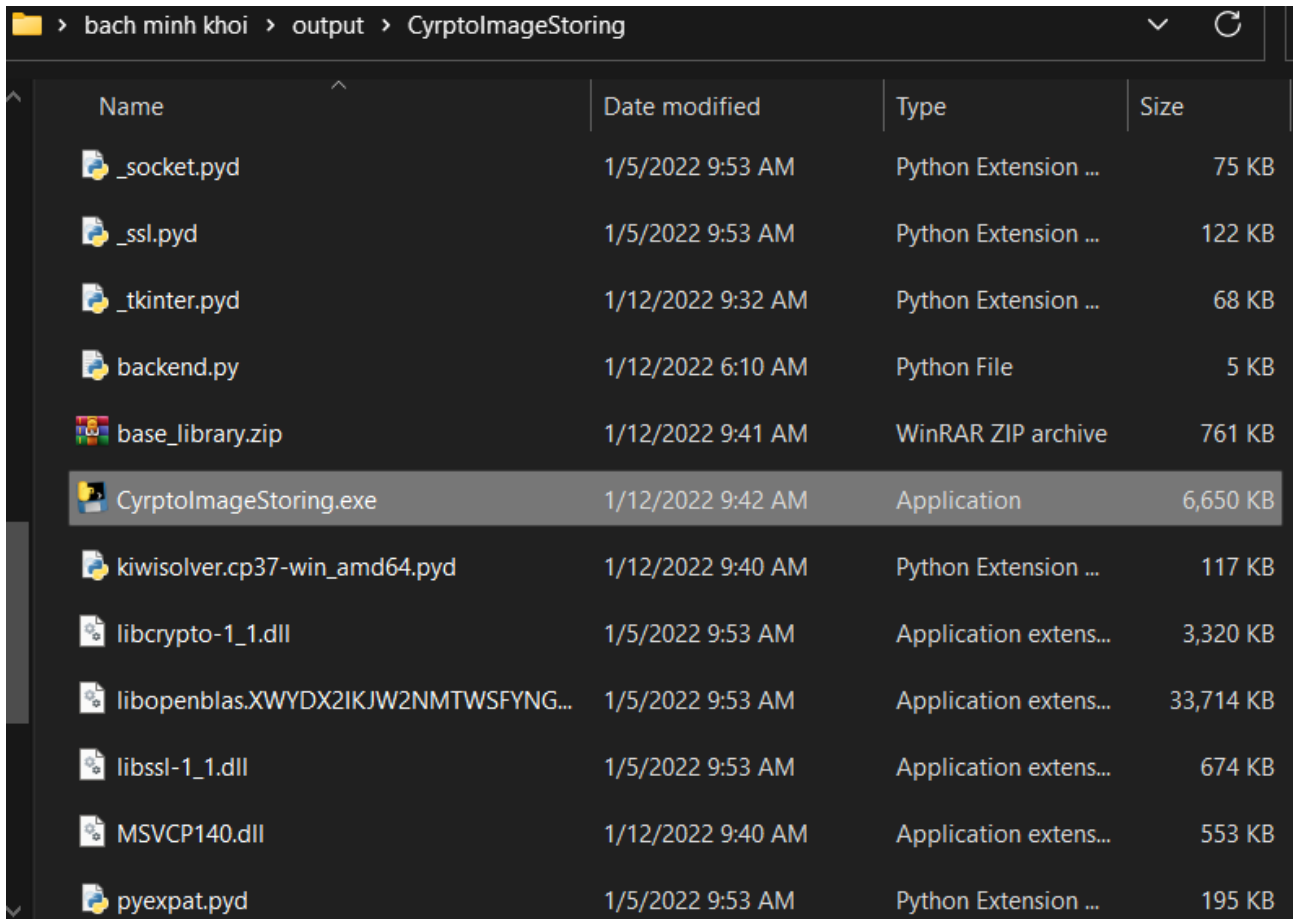
        image[i,j] = [M\_R, M\_G, M\_B]

    return image

## 4. HƯỚNG DẪN SỬ DỤNG CHƯƠNG TRÌNH

### 4.1. Sử dụng phần mềm được build sẵn (khuyến dùng)

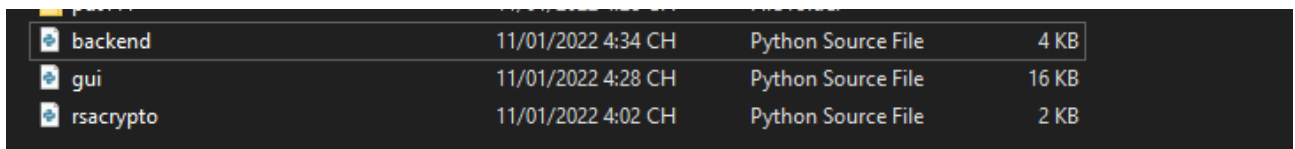
- B1: tải và giải nén chương trình để sử dụng
- B2: vào trong thư mục vừa giải nén, tìm file “**CryptolImageStoring.exe**” để khởi chạy và bắt đầu sử dụng chương trình



Name	Date modified	Type	Size
_socket.pyd	1/5/2022 9:53 AM	Python Extension ...	75 KB
_ssl.pyd	1/5/2022 9:53 AM	Python Extension ...	122 KB
_tkinter.pyd	1/12/2022 9:32 AM	Python Extension ...	68 KB
backend.py	1/12/2022 6:10 AM	Python File	5 KB
base_library.zip	1/12/2022 9:41 AM	WinRAR ZIP archive	761 KB
<b>CryptolImageStoring.exe</b>	1/12/2022 9:42 AM	Application	6,650 KB
kiwisolver.cp37-win_amd64.pyd	1/12/2022 9:40 AM	Python Extension ...	117 KB
libcrypto-1_1.dll	1/5/2022 9:53 AM	Application extens...	3,320 KB
libopenblas.XWYDX2IKJW2NMTWSFYNG...	1/5/2022 9:53 AM	Application extens...	33,714 KB
libssl-1_1.dll	1/5/2022 9:53 AM	Application extens...	674 KB
MSVCP140.dll	1/12/2022 9:40 AM	Application extens...	553 KB
pyexpat.pyd	1/5/2022 9:53 AM	Python Extension ...	195 KB

### 4.2. Biên dịch và chạy từ mã nguồn

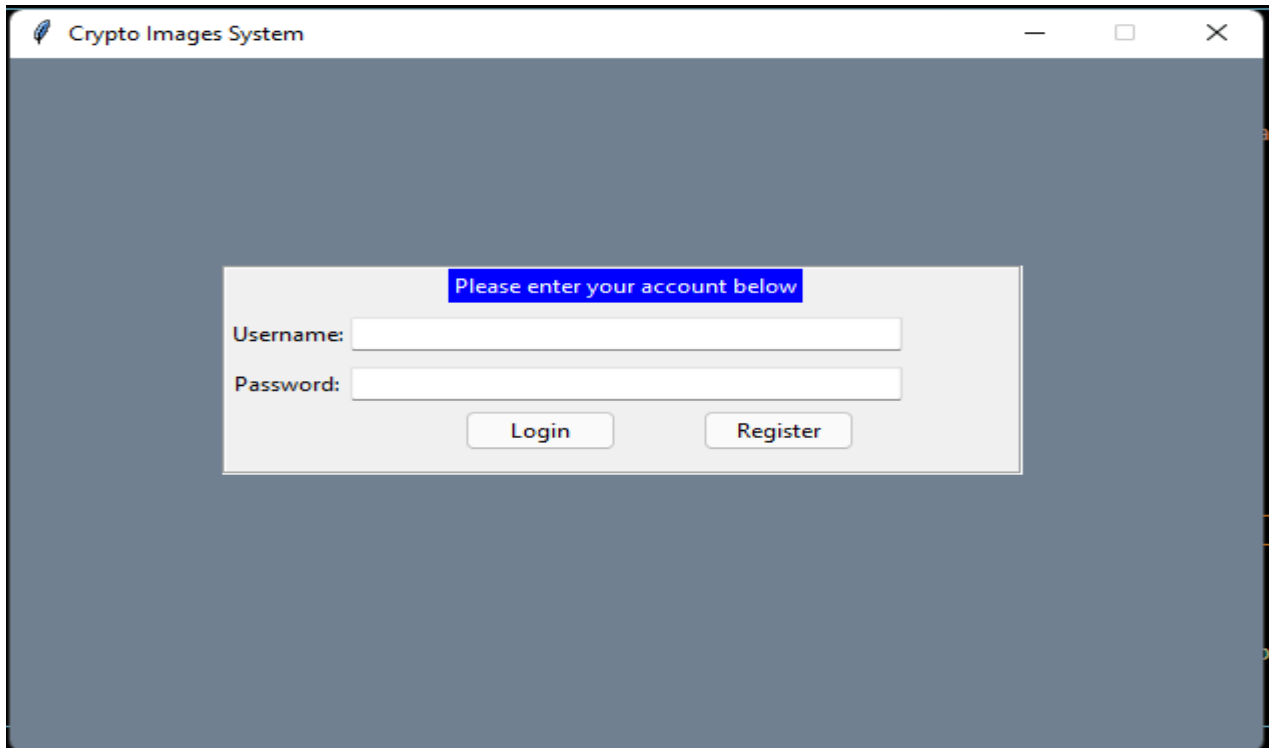
Mã nguồn gồm 3 file chính:



Name	Date modified	Type	Size
backend	11/01/2022 4:34 CH	Python Source File	4 KB
gui	11/01/2022 4:28 CH	Python Source File	16 KB
rsacrypto	11/01/2022 4:02 CH	Python Source File	2 KB

Để chạy chương trình, ta chạy file gui.py. Có thể chạy bằng các trình biên dịch (ví dụ: Visual Studio) có extension python.

Sau khi chạy thành công giao diện đầu tiên của chương trình sẽ như sau:



### 4.3. Tạo tài khoản

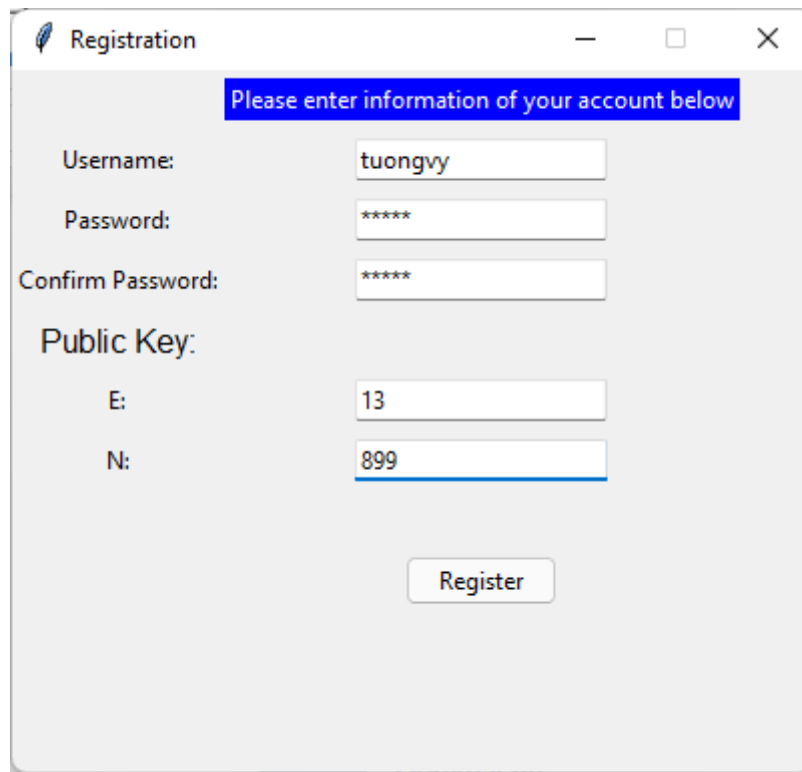
Tại màn hình đăng nhập của chương trình. Ta chọn Register để tạo tài khoản. Sau khi chọn Register chương trình sẽ có một cửa sổ đăng kí như sau:

A screenshot of a registration window titled "Registration". The window has a light gray background. At the top, a blue banner reads "Please enter information of your account below". Below the banner, there are four input fields: "Username:", "Password:", "Confirm Password:", and "Public Key:". The "Public Key:" field is expanded into two sub-fields: "E:" and "N:". At the bottom center of the window, there is a "Register" button.

Tại đây sẽ nhập đầy đủ username, password, confirm password (trùng với password) và khóa public RSA (E,N).

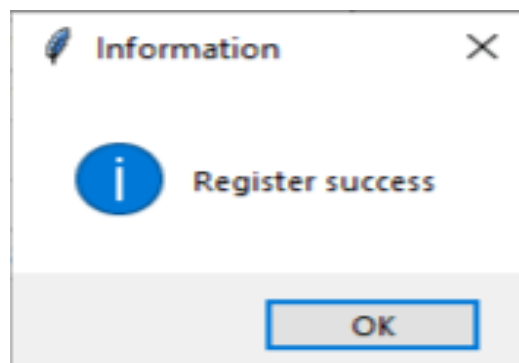


Ví dụ:

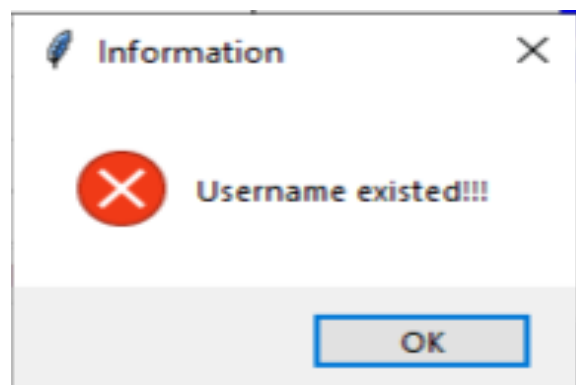


A screenshot of a 'Registration' window. At the top, a blue banner contains the text 'Please enter information of your account below'. Below this, there are four input fields: 'Username:' with the value 'tuongvy', 'Password:' with '\*\*\*\*\*', 'Confirm Password:' with '\*\*\*\*\*', and 'Public Key:'. The 'Public Key' section has two sub-fields: 'E:' with the value '13' and 'N:' with the value '899'. At the bottom center is a 'Register' button.

Đăng kí thành công màn hình sẽ có một thông báo như sau:



Nếu đăng kí thất bại (ví dụ còn chỗ chưa điền, tên đăng nhập đã tồn tại,v,v) màn hình sẽ như sau:

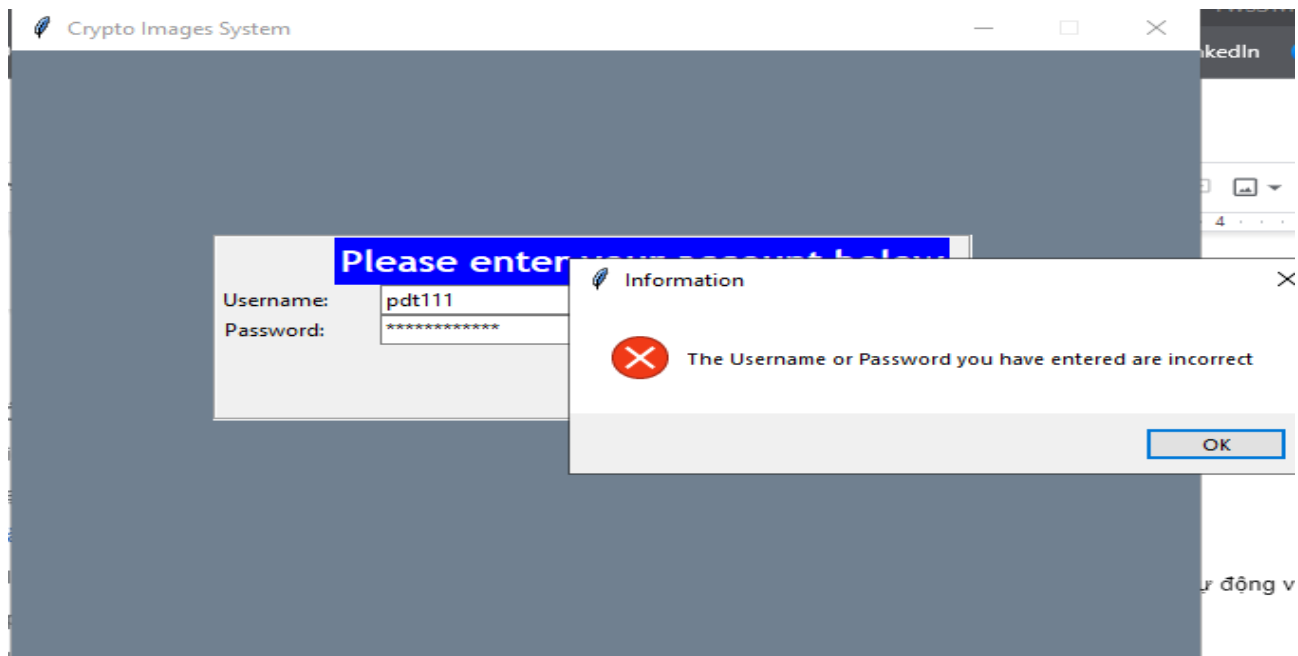


## 4.4. Đăng nhập

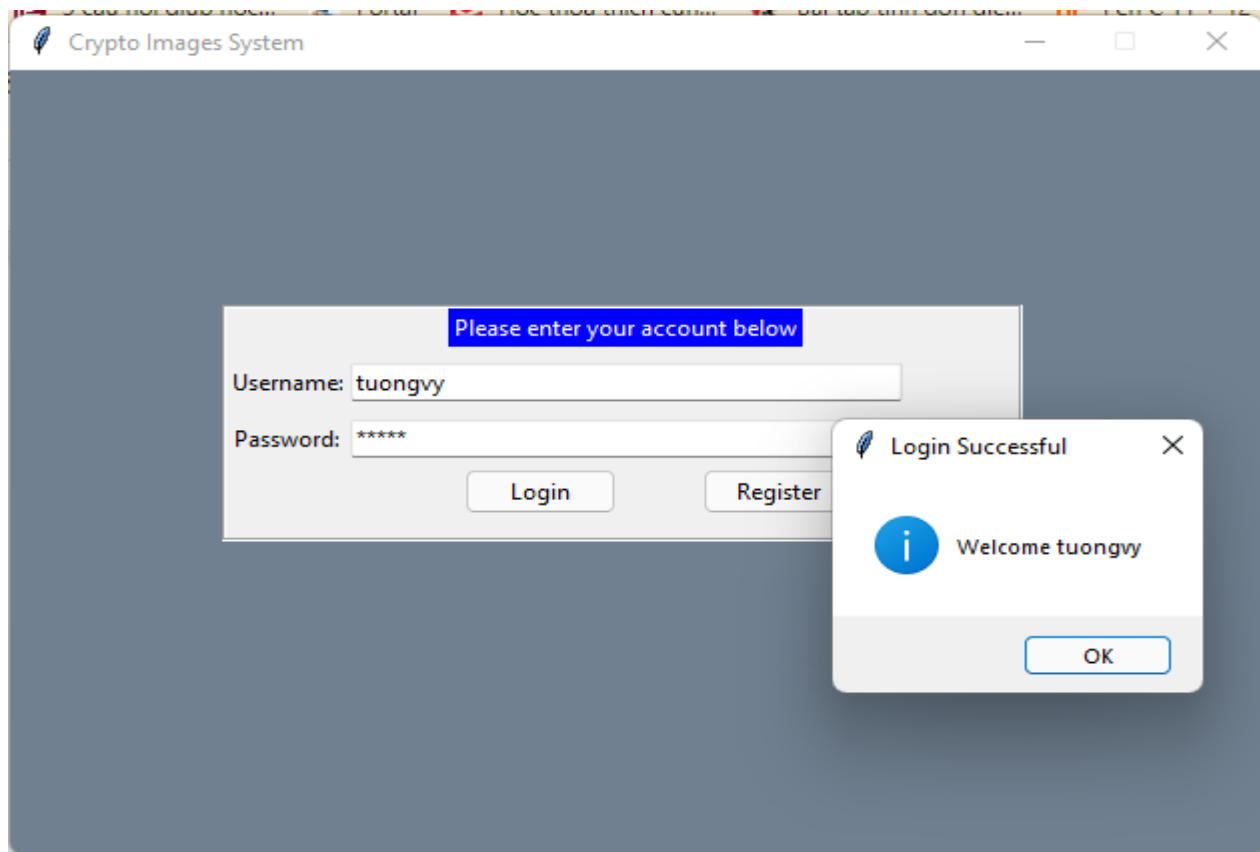
Sau khi đăng kí thành công chương trình sẽ tự động về màn hình đăng nhập:

Nhập tài khoản và mật khẩu vừa đăng kí.

Nếu đăng nhập sai hoặc còn chỗ chưa điền thì sẽ có màn hình lỗi như sau:

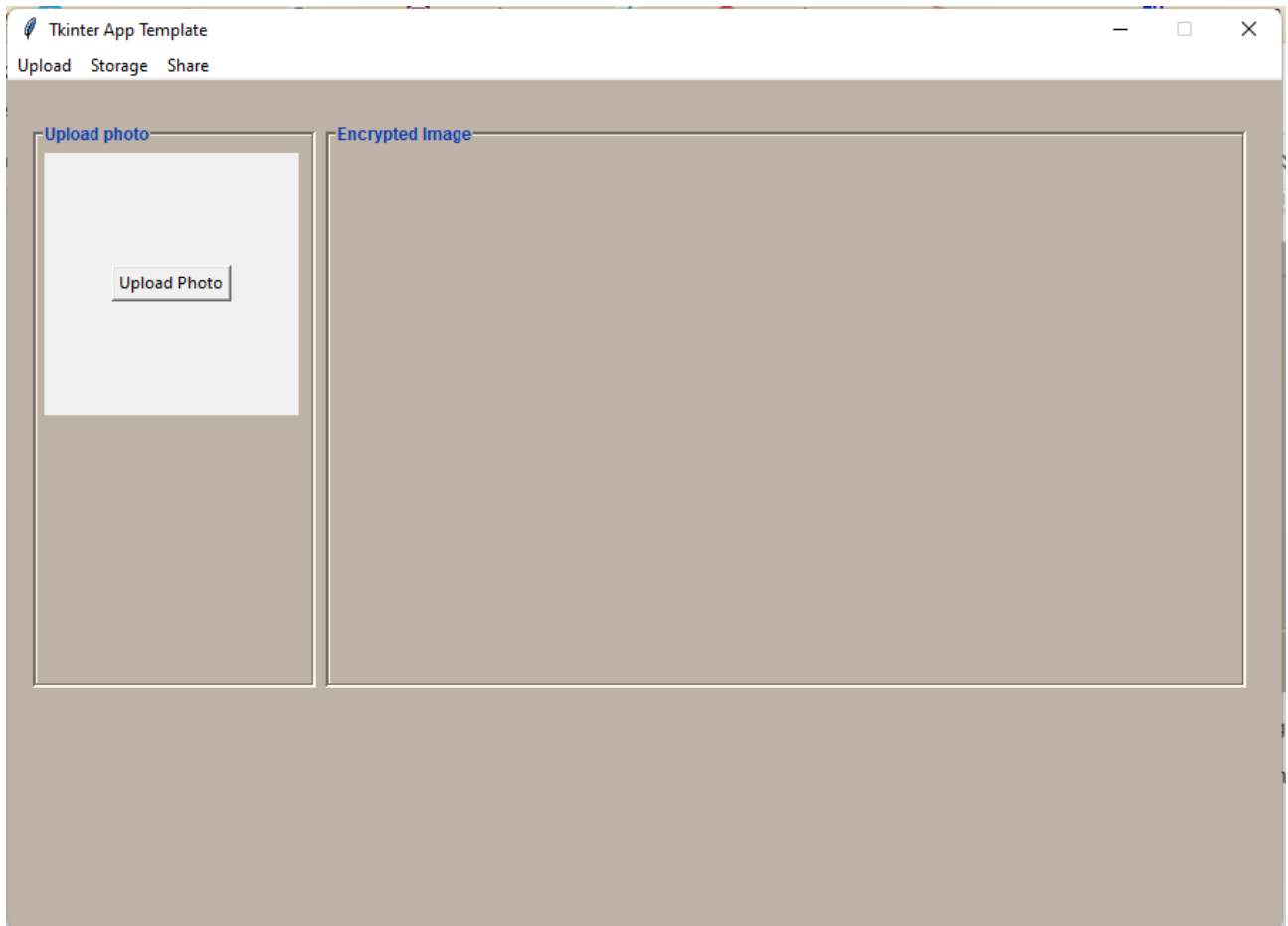


Đăng nhập thành công sẽ có thông báo như sau:



Và khi ấn ok sẽ vào màn hình chính của chương trình:

Menu bar ở màn hình chính sẽ bao gồm upload,storage,share



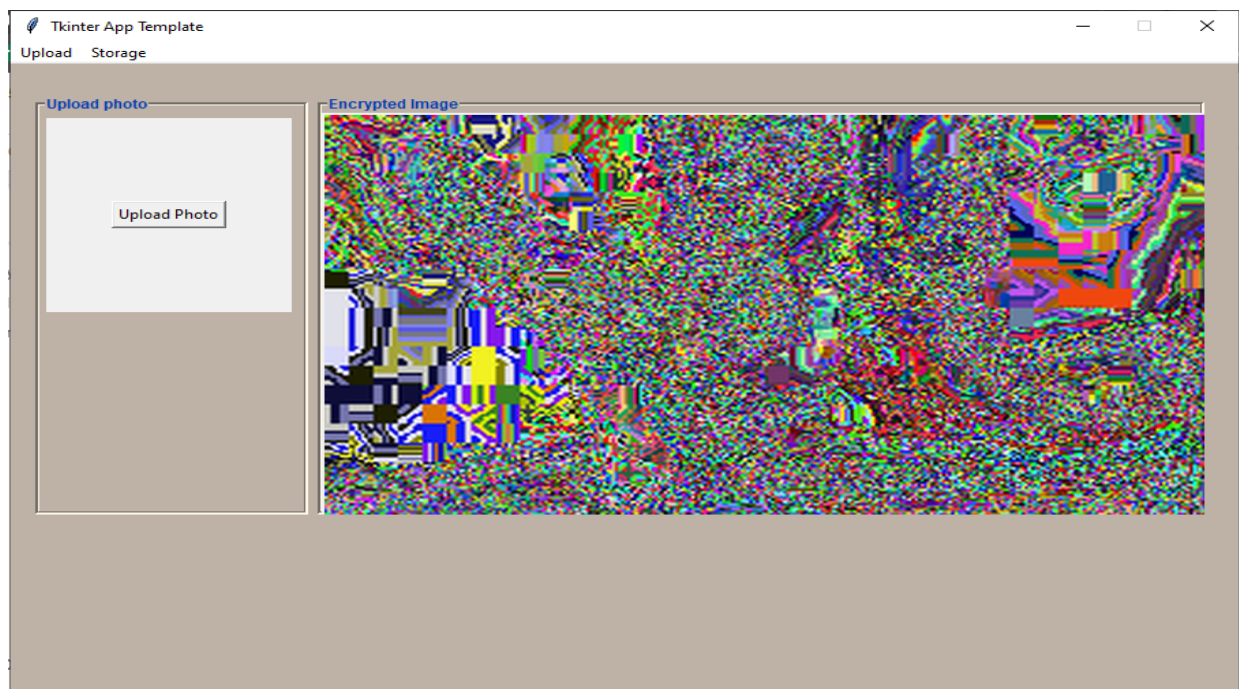
## 4.5. Upload ảnh lên server

Người dùng chọn upload photo. Chọn một ảnh muốn gửi lên server. Sau đó chương trình hoàn thành việc mã hóa và gửi lên server.



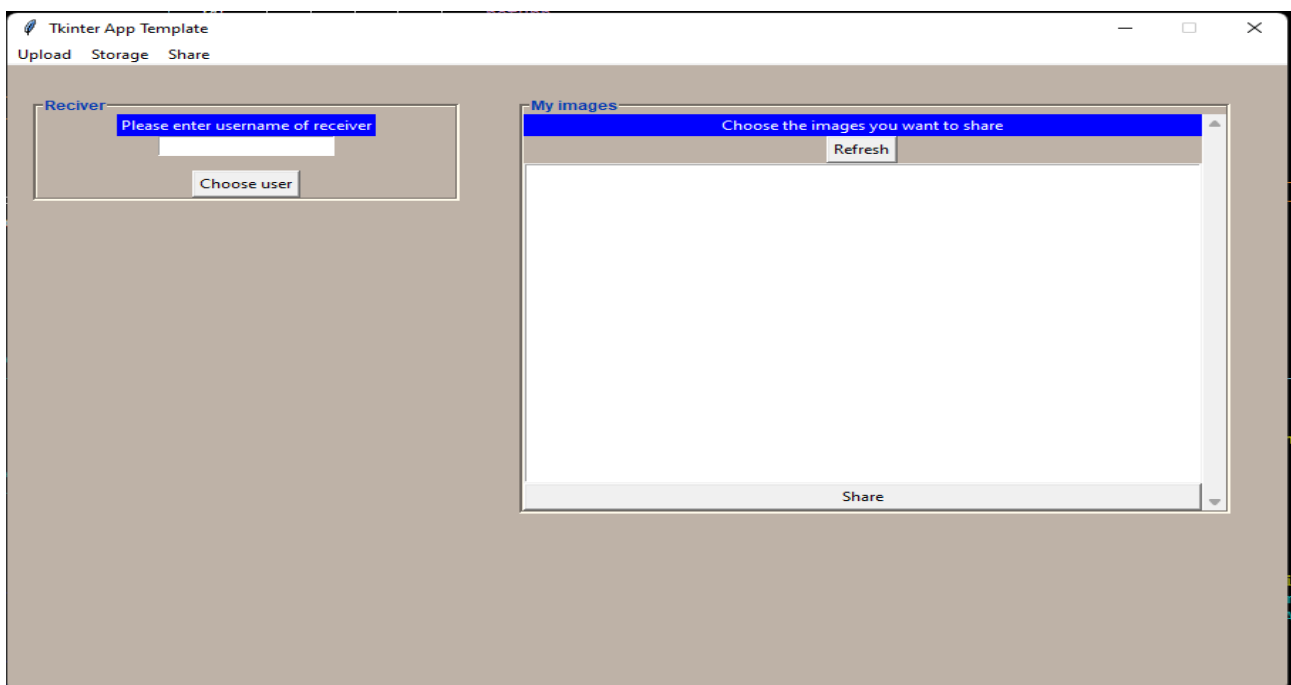
## 4.6. Mã hóa

Kết quả sau khi mã hóa:



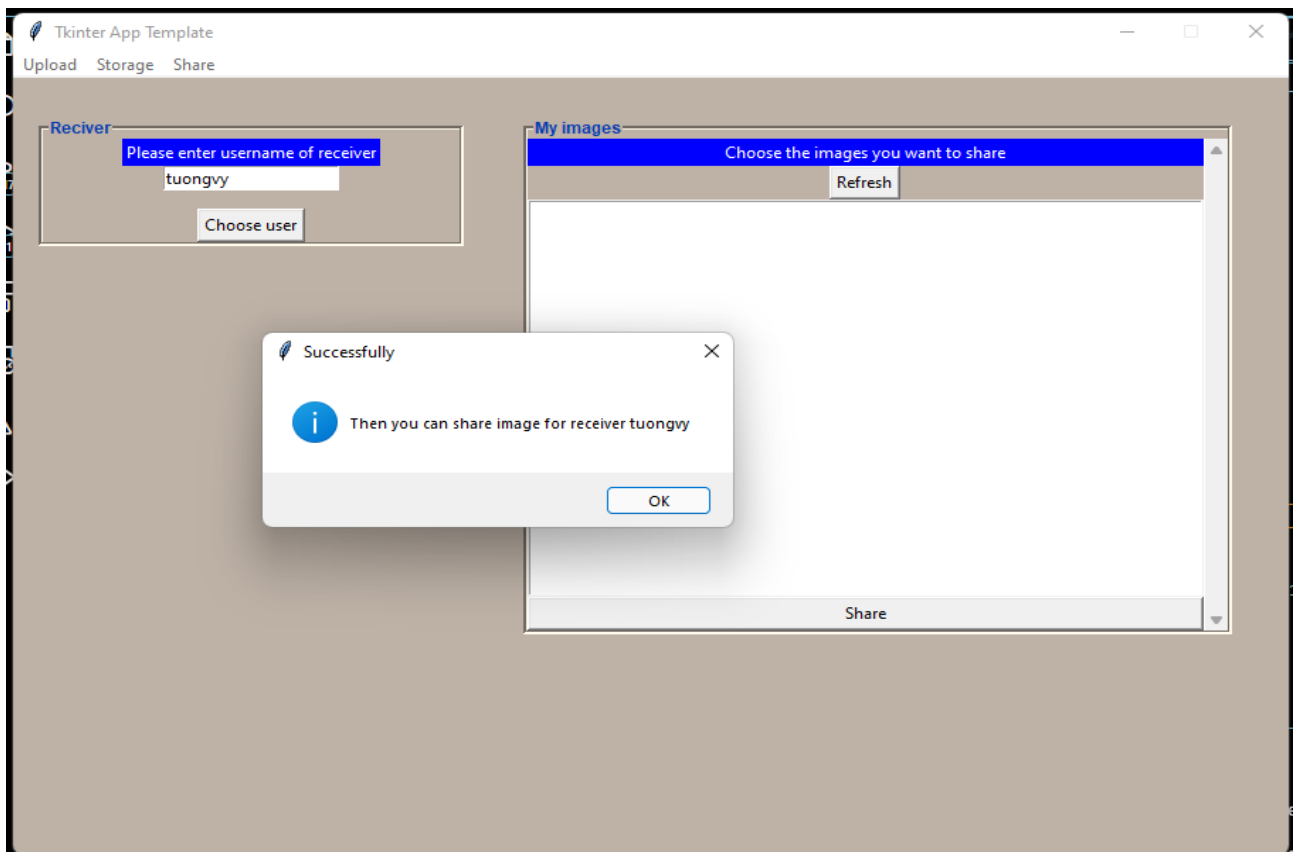
## 4.7. Chia sẻ ảnh

Khi nhấp vào Share trên thanh menu bar ở màn hình chính. Màn hình sử dụng cho việc chia sẻ ảnh sẽ hiển thị

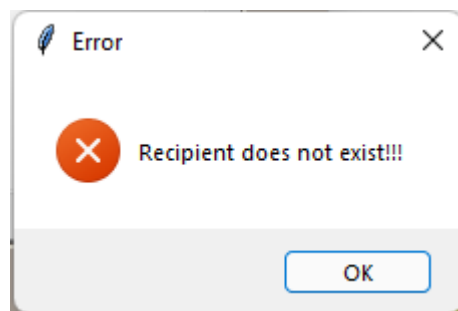


Nhập username của người cần chia sẻ ảnh ở khung Receiver

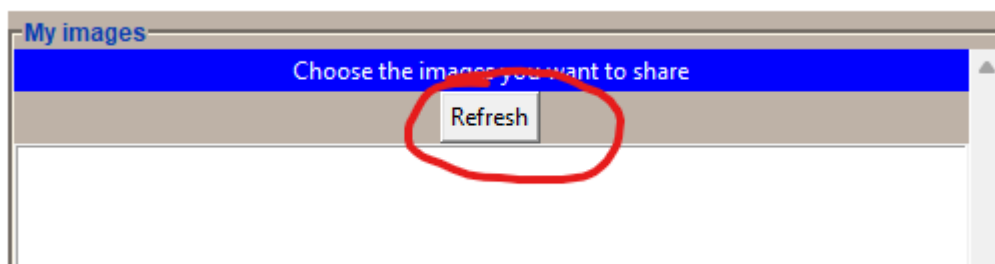
Nếu username của người nhận tồn tại trong hệ thống thì sẽ cho phép người dùng chia sẻ ảnh và hiển thị thông báo



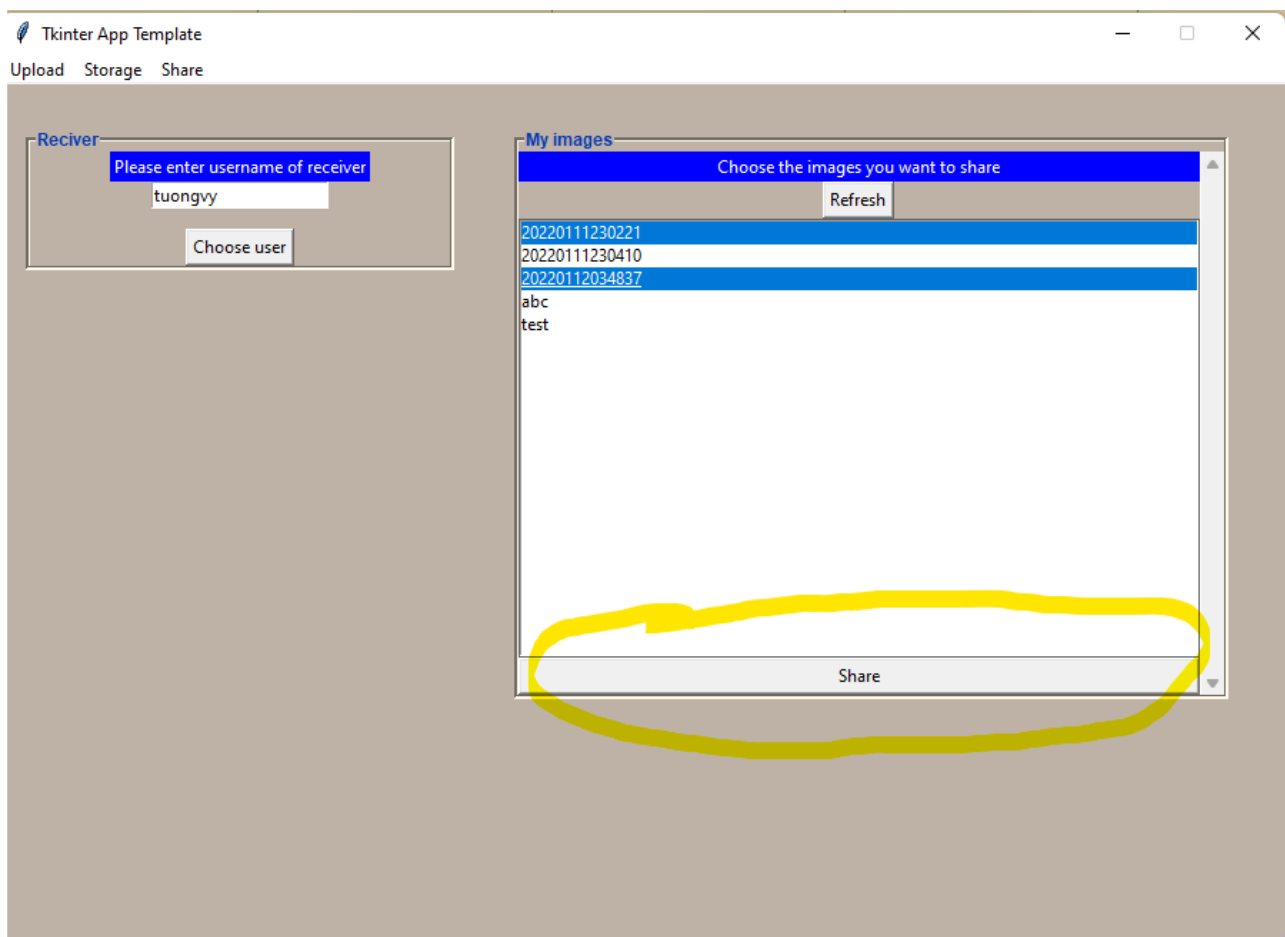
Ngược lại, nếu người nhận không tồn tại sẽ hiện thông báo:



Sau đó nhấn nút Refresh để hiển thị danh sách tệp hình ảnh của mình

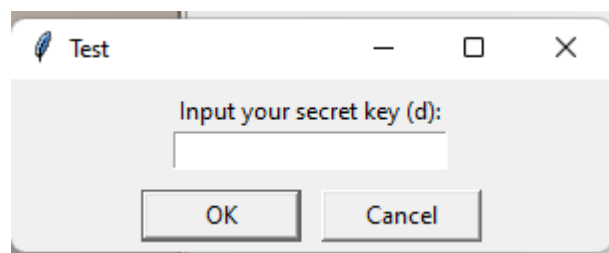


Lần lượt chọn các tệp ảnh muốn chia sẻ bằng cách nhấn vào tên các tệp:

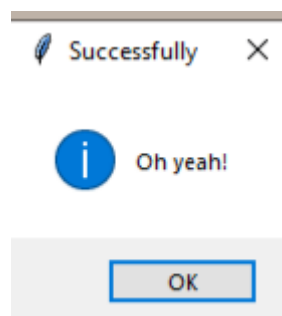


Sau đó nhấn nút Share , dialog sẽ hiện ra cho phép người gửi nhập private key của mình để giải mã ảnh.

Tiếp đó, lần lượt các ảnh cần chia sẻ sẽ được chia sẻ cho người nhận. Quá trình có thể diễn ra mất vài phút nếu số ảnh share lớn bởi vì ảnh sẽ được giải mã và mã hoá lại theo PUBLIC KEY của người nhận. Nhằm người được chia sẻ có thể tải ảnh rõ được chia sẻ.

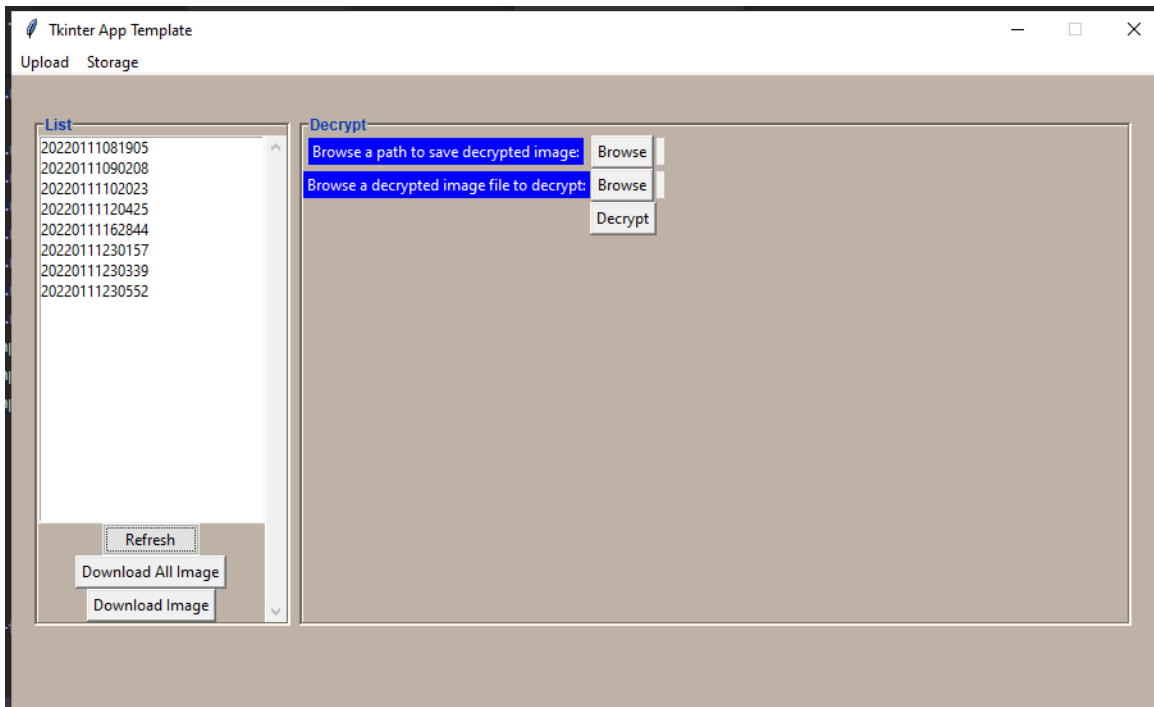


Quá trình nếu không xảy ra lỗi gián đoạn thì sau khi share được hình ảnh sẽ hiển thị thông báo



## 4.8. Tải ảnh

Để tải một ảnh. Ta vào phần Storage. Ấn refresh để xem danh sách các hình ảnh đã post lên server.

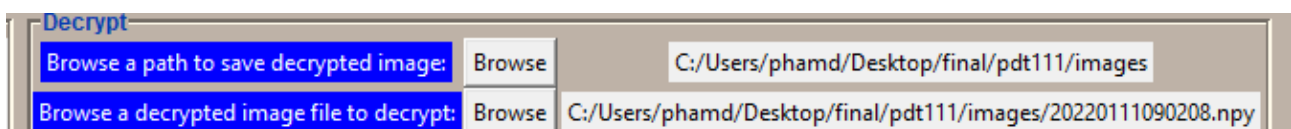


- Để tải ta chọn vào tên muốn tải và ấn Download Image. Chọn thư mục muốn lưu lại. Kết quả là một file .npy dùng để giải mã
- Để tải tất cả ta chọn Download All và chọn thư mục muốn lưu lại. Kết quả là một file zip chứa các file .npy để giải mã.

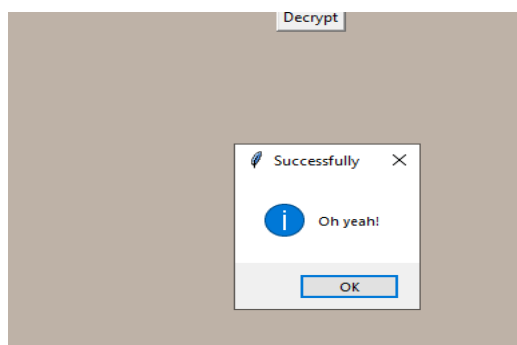
## 4.9. Giải mã

Ở bên trái ta có chức năng giải mã:

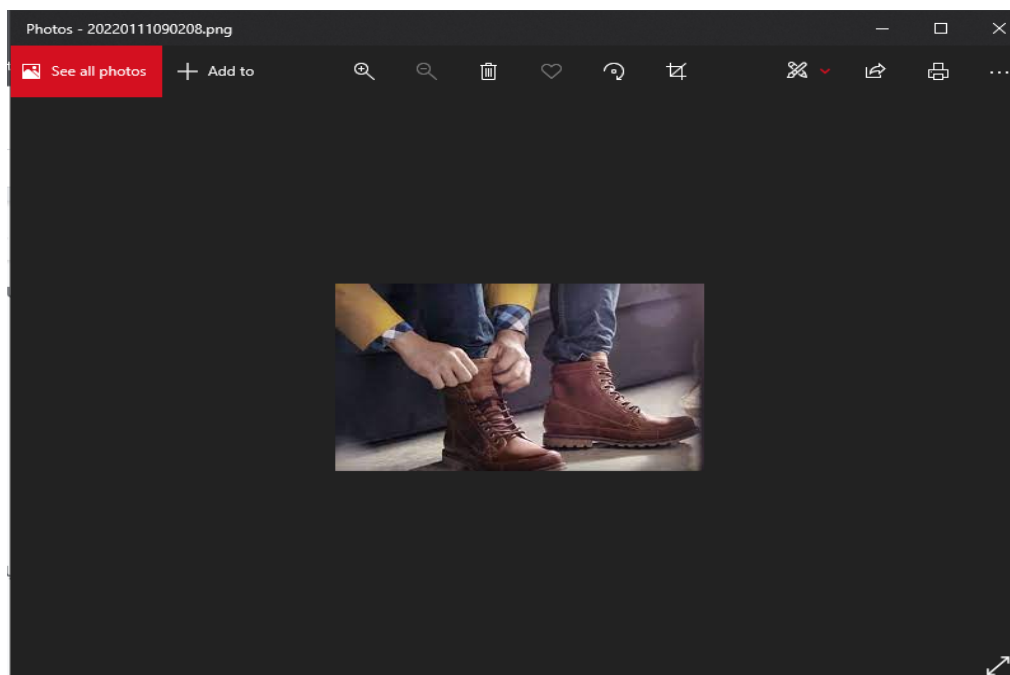
- Dòng đầu tiên là chọn đường dẫn muốn lưu ảnh được giải mã.
- Dòng thứ 2 chọn file .npy để giải mã hình ảnh.



- Ấn Decrypt và nhập mã bí mật để giải mã
- Đợi một lúc, nếu màn hình xuất hiện như vậy là thành công:



- Kết quả:





## 5. TỰ ĐÁNH GIÁ

### 5.1. Ưu điểm

Bảo mật mật khẩu (sử dụng bcrypt để mã hoá mật khẩu lưu vào database - tránh được tấn công vào database sẽ không làm mất dữ liệu người dùng).

Lưu trữ ảnh theo tên mà người dùng đặt và đúng định dạng ảnh.

Server chỉ lưu trữ dữ liệu đã được mã hoá và không bị lộ thông tin từ hình ảnh trên đường truyền giữa server - client nhờ vào việc mã hoá và giải mã trực tiếp trên client.

Server và database hoạt động độc lập và được deploy trên máy chủ Heroku, không cần chạy trên localhost. Có thể hoạt động như 1 hệ thống thật.

Ứng dụng cho user có thể dọn các file tạm/rác xuất hiện trong quá trình mã hoá giải mã sau khi tắt app.

Ứng dụng được build cho user đã đóng gói đầy đủ các môi trường cần thiết với mục đích có thể sử dụng ngay lập tức mà không cần phải cài đặt thêm package hay thư viện.

Ứng dụng cho phép người dùng tải ảnh đã mã hoá và ảnh bản rõ nếu nhập private key của bản thân.

### 5.2. Nhược điểm

Chưa phân vùng thư mục lưu trữ ảnh trên server, có khả năng bị xung đột khi có ít nhất 2 hoặc nhiều người dùng cùng gửi yêu cầu đến server với ảnh cùng tên (do các ảnh được đặt cùng trong 1 folder - do việc tạo directory trên local Heroku là không khả thi mà cần thêm extension, tuy nhiên chưa thực hiện được do giới hạn nhân lực).

Chưa cung cấp đủ các chức năng nâng cao cho hệ thống ( đổi mật khẩu, xóa tài khoản, xóa hình, hủy chia sẻ, đổi tên ảnh)

Tốc độ xử lý mã hoá giải mã còn chậm (do mã hoá toàn bộ khung ảnh và sử dụng thuần RSA, nếu kết hợp với Hill cipher thì tốc độ sẽ tăng nhanh rất nhiều lần).

Chưa có kiểm soát số lượng và quyền hạn truy cập, attacker có thể tấn công server (DDOS, ...) hoặc lấy ảnh được mã hoá của user khác nếu biết username và tên ảnh.

Giao diện với tkinter chưa thật sự tốt.

Việc chia sẻ ảnh cho người dùng khác diễn ra mất vài phút và có thể lâu hơn nếu số lượng ảnh lớn. Với vì ảnh mã hoá được lấy từ firebase được server gửi về cần phải được giải mã theo public key của người gửi và mã hoá lại theo private key của người nhận. Nhằm giúp cho người nhận có thể tải ảnh bản rõ của ảnh được chia sẻ.

### 5.3. Mở rộng

Bảo mật cơ sở dữ liệu.

Mã hoá thông tin truyền gửi giữa server - client.

Phân quyền người dùng (cơ sở dữ liệu).

Thiết lập kết nối bảo mật hơn (yêu cầu authentication với mỗi request, tránh attack lấy ảnh từ kho của user khác mà chưa được chia sẻ).

## 6. TÀI LIỆU THAM KHẢO

[Welcome to Flask — Flask Documentation \(2.0.x\) \(palletsprojects.com\)](#)

[Python REST API Tutorial - Building a Flask REST API - YouTube](#)

[Request Parsing — Flask-RESTful 0.3.8 documentation](#)

[How to Handle Request JSON Data in Flask - YouTube](#)

[HTTP response status codes - HTTP | MDN \(mozilla.org\)](#)

[REST Client - Visual Studio Marketplace](#)

[How to Encrypt a Password in Python Using bcrypt \(makeuseof.com\)](#)

[Receive or Return files Flask Python | Analytics Vidhya \(medium.com\)](#)

[python - Send with multiple CSVs using Flask? - Stack Overflow](#)

[python - Get the data received in a Flask request - Stack Overflow](#)

[Handling File Uploads With Flask - miguelgrinberg.com](#)

[flask.request.files.getlist Example \(programtalk.com\)](#)

[Listing All Your Files From Firebase Storage 🔥 | Retrieving Data From Firebase Cloud Storage - YouTube](#)

[python - Flask to Numpy Image Conversion - Stack Overflow](#)

[Receive and Send back Image in Flask: In memory solution | by RAJAT KANTI Bhattacharjee | csmadeeasy | Medium](#)

[gjpamv12n4\\_73.pdf \(ripublication.com\)](#)

[Python GUI Programming With Tkinter – Real Python](#)