

Bộ Giáo Dục Và Đào Tạo  
Trường Đại Học Ngoại Ngữ - Tin Học Thành Phố Hồ Chí Minh  
**Khoa Công Nghệ Thông Tin**



## **BÁO CÁO KẾT THÚC HỌC PHẦN LẬP TRÌNH MẠNG**

**ĐỀ TÀI : Xây dựng chương trình nhận diện khuôn  
mặt**

**Giáo Viên Hướng Dẫn : ThS Phạm Phú Thiện**

**Thành Viên :**

1. Phạm Hoàng Gia Bảo – MSSV: 22DH110298
2. Lê Thành Đạt – MSSV: 22DH110717
3. Huỳnh Minh Nhựt – MSSV: 22DH112633
4. Dương Lê Huy Hoàng– MSSV: 22DH114536

*Tp. Hồ chí minh, Ngày 24 tháng 07 năm 2024*

### Nhận xét của giảng viên

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

## Lời cảm ơn

Để hoàn thành tốt bài báo cáo này, chúng em xin gửi lời cảm ơn chân thành đến giảng viên, Ts Phạm Phú Thiện, người đã hỗ trợ cho chúng em trong quá trình làm bài. Cảm ơn thầy đã đưa ra những góp ý, nhận xét để chúng em có thể hoàn thành tốt bài báo cáo này và nộp đúng hạn thời gian bài báo cáo đề ra.

Trong thời gian học tập và làm báo cáo dưới sự hướng dẫn của thầy, chúng em đã có thêm những kiến thức bổ ích, những kinh nghiệm để có thể đọc được các biểu đồ và tự tay mình viết được một báo cáo về môn hệ điều hành mạng.

Mặc dù đã nỗ lực trong việc hoàn thiện báo cáo, nhưng do thời gian có hạn, bước đầu đi vào thực tế để tìm hiểu và triển khai một công cụ theo dõi và giám sát trên linux trong một thời gian có hạn, với lượng kiến thức nông cạn và hạn chế, nhiều chỗ khi làm một thứ mới mẻ mình chưa bao giờ được làm, nên bài báo cáo về đề tài: “Xây dựng chương trình nhận diện khuôn mặt” của chúng em chắc chắn vẫn còn rất nhiều sai sót nên chúng em rất mong rằng mình có thể nhận được những lời góp ý quý báu của các thầy cô để chúng em có thể hoàn thiện kiến thức của bản thân mình hơn trong việc xây dựng hệ thống client – server và chúng em có thể làm tốt hơn nữa trong những đề án lần sau.

Một lần nữa chúng em xin chân thành cảm ơn thầy và luôn mong nhận sự đóng góp của quý thầy cô.

Cuối lời, chúng em xin kính chúc quý thầy cô Khoa Công Nghệ Thông Tin luôn dồi dào sức khỏe và thành công hơn nữa trong sự nghiệp trồng người của mình.

Chúng em trân trọng cảm ơn!

## Mục Lục

<b>Nhận xét của giảng viên .....</b>	<b>2</b>
<b>Lời cảm ơn .....</b>	<b>3</b>
<b>Mục Lục.....</b>	<b>4</b>
<b>Danh Mục Bảng Biểu .....</b>	<b>6</b>
<b>Danh Mục Hình Ảnh.....</b>	<b>7</b>
<b>I. Tổng quan về đề án .....</b>	<b>8</b>
1. Giới thiệu lý do chọn đề tài .....	8
2. Tầm quan trọng của đề tài trong thực tế.....	8
3. Mục tiêu nghiên cứu của đề tài.....	8
4. Phạm vi của đề tài.....	8
<b>II. Phân tích yêu cầu .....</b>	<b>8</b>
1. Yêu cầu chức năng .....	8
2. Yêu cầu phi chức năng .....	9
3. Các kịch bản sử dụng (use case) .....	9
<b>III. Thiết kế hệ thống.....</b>	<b>15</b>
1. Kiến trúc hệ thống .....	15
2. Thiết kế chi tiết.....	17
Thiết kế giao diện người dùng.....	17
Thiết kế cơ sở dữ liệu .....	23
Thiết kế các module và các luồng dữ liệu.....	24
<b>IV. Mã nguồn .....</b>	<b>25</b>
1. Môi trường phát triển.....	25
2. Cấu trúc mã nguồn .....	25
3. Quá trình cài đặt và triển khai hệ thống .....	27
4. Các đoạn mã nguồn quan trọng .....	37

5. Mô tả các điểm nổi bật trong mã nguồn .....	39
<b>V. Đánh giá và kết luận.....</b>	<b>39</b>
1. Kết quả thực hiện.....	39
2. Đánh giá hệ thống .....	40
3. Kết luận.....	40
4. Bảng phân công .....	40
<b>VI. Tài liệu tham khảo .....</b>	<b>41</b>

## **Danh Mục Bảng Biểu**

<b>Bảng 1. Use Case - Chụp ảnh .....</b>	<b>10</b>
<b>Bảng 2. Use Case - Chọn ảnh từ thư mục .....</b>	<b>11</b>
<b>Bảng 3. Use Case - Thêm dữ liệu .....</b>	<b>12</b>
<b>Bảng 4. Use Case - Nhận diện khuôn mặt.....</b>	<b>13</b>
<b>Bảng 5. Use Case - Ghi dữ liệu vào database.....</b>	<b>14</b>
<b>Bảng 6. Bảng phân công công việc.....</b>	<b>40</b>

## Danh Mục Hình Ảnh

Hình 1. Use case tổng quát.....	10
Hình 2. Cấu trúc client.....	16
Hình 3. Cấu trúc Server.....	17
Hình 4. Trang chủ client.....	18
Hình 5. Trang thêm dữ liệu.....	19
Hình 6. Trang kết quả sau khi thêm dữ liệu.....	20
Hình 7. Trang nhận diện bằng camera và kết quả .....	21
Hình 8. Trang chọn ảnh tĩnh để gửi .....	22
Hình 9. Trang kết quả nhận diện sau khi gửi ảnh tĩnh .....	23
Hình 10. Tạo cơ sở dữ liệu.....	23
Hình 11. ERD cơ sở dữ liệu .....	24
Hình 12. Cấu trúc client.....	26
Hình 13. Cấu trúc Server.....	27
Hình 14. Thư mục đồ án .....	27
Hình 15. Mở project lên.....	28
Hình 16. Cài đặt các file jar này trên mạng về máy .....	28
Hình 17. Thêm thư viện openCV .....	29
Hình 18. Tạo cơ sở dữ liệu.....	30
Hình 19. Chọn kết nối mới.....	31
Hình 20. Chọn kiểu driver.....	32
Hình 21. Chọn file jar connect .....	33
Hình 22. Nhập password và finish .....	34
Hình 23. Chạy giao diện.....	36
Hình 24. Trang chủ .....	36
Hình 25. Nhập thông tin và gửi.....	36
Hình 26. Nhận diện khuôn mặt camera .....	37
Hình 27. Đoạn mã nhận dữ liệu và xử lý.....	37
Hình 28. Lưu hình ảnh và dữ liệu vào cơ sở dữ liệu.....	38
Hình 29. Nhận diện khuôn mặt.....	39
Hình 30. Project Plan 1 .....	41
Hình 31. Project Plan 2.....	41

## **I. Tổng quan về đề án**

### **1. Giới thiệu lý do chọn đề tài**

**Lý do chọn đề tài:** Trong thời đại công nghệ hiện nay, nhận diện khuôn mặt và đối tượng đã trở thành một phần quan trọng trong nhiều ứng dụng, từ bảo mật, quản lý nhân sự đến phân tích hình ảnh. Công nghệ nhận diện giúp cải thiện khả năng tự động hóa và nâng cao sự tiện lợi trong nhiều lĩnh vực.

### **2. Tầm quan trọng của đề tài trong thực tế**

**Tầm quan trọng của đề tài:** Đề tài này có thể đóng góp vào việc phát triển các ứng dụng thực tiễn như hệ thống giám sát an ninh, xác thực người dùng và các công nghệ thông minh khác. Nó giúp tăng cường bảo mật và hiệu quả của các hệ thống nhận diện.

### **3. Mục tiêu nghiên cứu của đề tài**

Mục tiêu chính của đề án này là xây dựng một chương trình nhận diện khuôn mặt và đối tượng với các chức năng sau:

- **Nhận diện khuôn mặt:**

- Gửi ảnh khuôn mặt từ webcam hoặc ảnh tĩnh lên server.
- Nhận kết quả trả về từ server, bao gồm tên hoặc thông tin cá nhân của người được nhận diện.
- Xác định mức độ khớp giữa ảnh đầu vào và các đối tượng trong cơ sở dữ liệu.
- Bổ sung khuôn mặt mới vào cơ sở dữ liệu.

- **Nhận diện đối tượng:**

- Gửi ảnh tĩnh lên server.
- Nhận kết quả trả về từ server, bao gồm danh sách các đối tượng được nhận diện trong ảnh.

### **4. Phạm vi của đề tài**

Đề án tập trung vào việc phát triển chương trình nhận diện khuôn mặt và đối tượng với giao diện người dùng đơn giản, dễ sử dụng. Chức năng nhận diện khuôn mặt sẽ sử dụng cơ sở dữ liệu ảnh khuôn mặt được cung cấp sẵn, trong khi chức năng nhận diện đối tượng sẽ sử dụng các thuật toán học máy để phân tích ảnh và xác định các đối tượng.

## **II. Phân tích yêu cầu**

### **1. Yêu cầu chức năng**



### **Yêu cầu về chức năng phía client (phải có GUI):**

- Đối với chức năng nhận diện khuôn mặt:
  - Gửi dữ liệu ảnh về server: client có thể gửi dữ liệu khuôn mặt của 1 đối tượng về server bằng 1 trong 2 phương pháp dưới đây (nhóm phải cài đặt cả 2 phương pháp):
    - Sử dụng webcam của client.
    - Sử dụng ảnh tĩnh có sẵn trên thiết bị.
  - Nhận kết quả phản hồi từ server: in ra đối tượng ứng với khuôn mặt nhận diện được. Trường hợp nhận diện được không nhận diện được, server cần trả về kết quả tỉ lệ % hoặc thông số cho biết mức độ khớp giữa ảnh đầu vào và các đối tượng trên CSDL. CSDL trên server phải có tối thiểu 5 đối tượng.
  - Bổ sung khuôn mặt: vẫn sử dụng hai phương pháp truyền bằng hình ảnh tĩnh hoặc lấy ảnh trực tiếp từ webcam.
- Đối với chức năng nhận diện đối tượng (object):
  - Client chỉ gửi dữ liệu ảnh tĩnh về server, server phân tích ảnh và trả kết quả các đối tượng xuất hiện trong ảnh cho client.

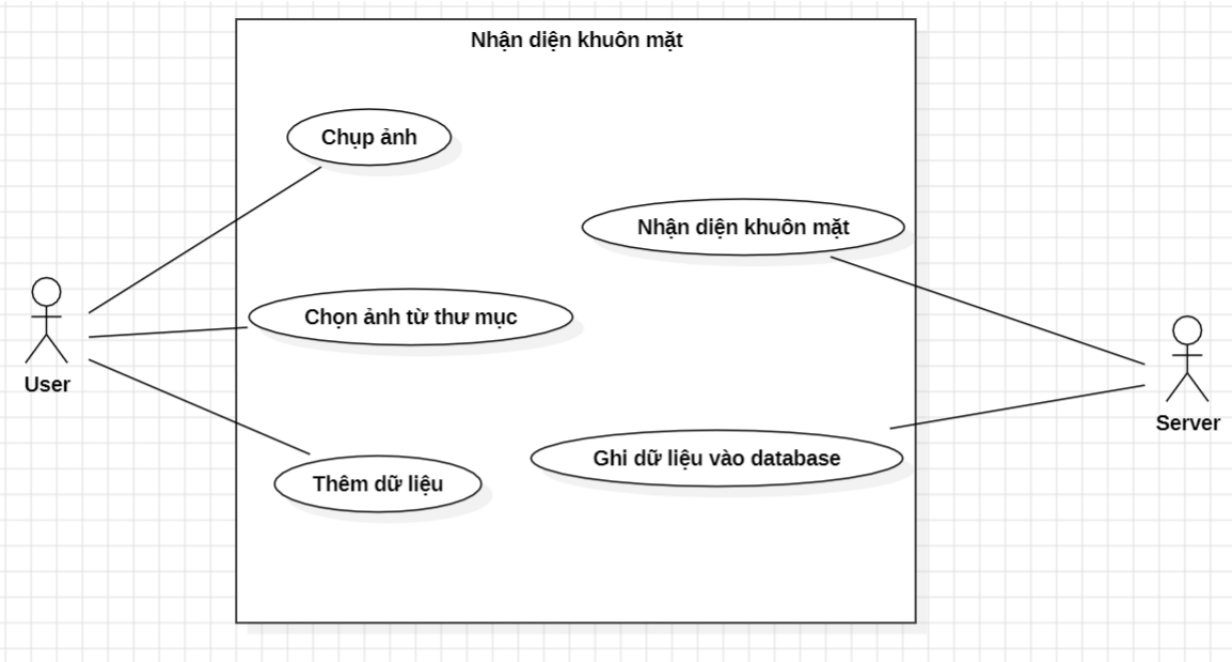
### **Yêu cầu về chức năng phía server (không cần GUI):**

- Server lưu dữ liệu mẫu của một số người dùng (tối thiểu là 5 người khác nhau, chưa kể các người dùng mới sẽ được thêm và test trong quá trình chấm bài), sử dụng cho việc nhận diện.
- Nhóm SV có thể sử dụng các thư viện/API/ngôn ngữ để giải quyết vấn đề phía server. Tuy nhiên, giao tiếp giữa client và server bắt buộc phải thông qua Java socket

## **2. Yêu cầu phi chức năng**

- **Hiệu năng:** Hệ thống cần xử lý ảnh và trả về kết quả nhanh chóng.
- **Bảo mật:** Dữ liệu truyền tải cần được bảo vệ để ngăn chặn truy cập trái phép.
- **Độ tin cậy:** Hệ thống cần hoạt động ổn định và chính xác trong các tình huống thực tế.

## **3. Các kịch bản sử dụng (use case)**



Hình 1. Use case tổng quát

Mô tả Use Case:

Bảng 1. Use Case - Chụp ảnh

Name	Chụp ảnh
Code	UC01
Short description	UC cho phép user chụp ảnh từ camera
Actor	User
Trigger	Actor nhấn nút camera
Pre-condition	Actor có camera
Post-Condition	Mở trang camera
Standard process(flow)	1. Actor nhấn nút camera 2. Hệ thống hiện thị giao diện camera 3. Actor chỉnh camera sao cho phù hợp 4. Actor nhấn nút camera capture 5. Hệ thống gửi hình ảnh đến server thông qua java socket

	6. Server nhận hình ảnh 7. Server xử lý dữ liệu và trả về kết quả 8. Hệ thống hiện thị thông tin người dùng gần giống nhất
<b>Alternative process</b>	
<b>Error situations</b>	1. Không thể kết nối được với server 2. Không kết nối internet.
<b>System state in error situations</b>	Không thể kết nối tới camera của bạn

**Bảng 2. Use Case - Chọn ảnh từ thư mục**

<b>Name</b>	Chọn ảnh từ thư mục
<b>Code</b>	UC02
<b>Short description</b>	UC cho phép user chọn ảnh từ thư mục
<b>Actor</b>	User
<b>Trigger</b>	Actor nhấn nút image
<b>Pre-condition</b>	Actor có ảnh trong thư mục
<b>Post-Condition</b>	Mở trang image
<b>Standard process(flow)</b>	1. Actor nhấn nút image 2. Hệ thống hiện thị trang image 3. Actor chọn nút select image 4. Actor chọn hình ảnh muốn nhận diện 5. Actor nhấn open 6. Hệ thống hiện thị hình ảnh đã chọn nếu đúng định dạng 7. Actor nhấn send image 8. Hệ thống gửi ảnh đến server

	9. Server nhận hình ảnh 10. Server nhận diện và trả về kết quả 11. Hệ thống hiển thị thông tin người dùng gần giống nhất
<b>Alternative process</b>	
<b>Error situations</b>	1. Không thể kết nối được với server 2. Không kết nối internet.
<b>System state in error situations</b>	

*Bảng 3. Use Case - Thêm dữ liệu*

<b>Name</b>	Thêm dữ liệu
<b>Code</b>	UC03
<b>Short description</b>	UC cho phép user thêm dữ liệu
<b>Actor</b>	User
<b>Trigger</b>	Actor nhấn nút add user
<b>Pre-condition</b>	
<b>Post-Condition</b>	Mở trang add user
<b>Standard process(flow)</b>	1. Actor nhấn nút add user 2. Hệ thống hiển thị form điền thông tin 3. Actor nhấn gửi 4. Hệ thống gửi các thông tin đến server 5. Server nhận thông tin và kiểm tra định dạng 6. Server lưu vào cơ sở dữ liệu

	7. Server thông báo lại cho hệ thống nếu lưu thành công 8. Hệ thống hiện thị đã lưu thành công
<b>Alternative process</b>	6'. Server thông báo sai định dạng 7'. Hệ thống hiện thị sai định dạng
<b>Error situations</b>	1. Không thể kết nối được với server 2. Không kết nối internet.
<b>System state in error situations</b>	

**Bảng 4. Use Case - Nhận diện khuôn mặt**

<b>Name</b>	Nhận diện khuôn mặt
<b>Code</b>	UC04
<b>Short description</b>	UC cho phép server nhận diện khuôn mặt
<b>Actor</b>	Server
<b>Trigger</b>	Actor nhận được hình ảnh và thông báo findFace
<b>Pre-condition</b>	
<b>Post-Condition</b>	Hiện thông tin người được nhận diện
<b>Standard process(flow)</b>	1. Actor nhận được thông báo findFace 2. Actor nhận hình ảnh 3. Actor so sánh hình ảnh nhận được với các hình ảnh trong cơ sở dữ liệu 4. Actor lưu lại đường dẫn hình ảnh giống nhất

	5. Actor kiểm tra đường dẫn này có thông tin người dùng thế nào trong database 6. Actor lấy các thông tin đó lưu lại 7. Actor gửi các thông tin được lưu lại về cho hệ thống 8. Hệ thống hiển thị thông tin của người trong ảnh.
<b>Alternative process</b>	
<b>Error situations</b>	1. Không thể kết nối được với server 2. Không kết nối internet.
<b>System state in error situations</b>	

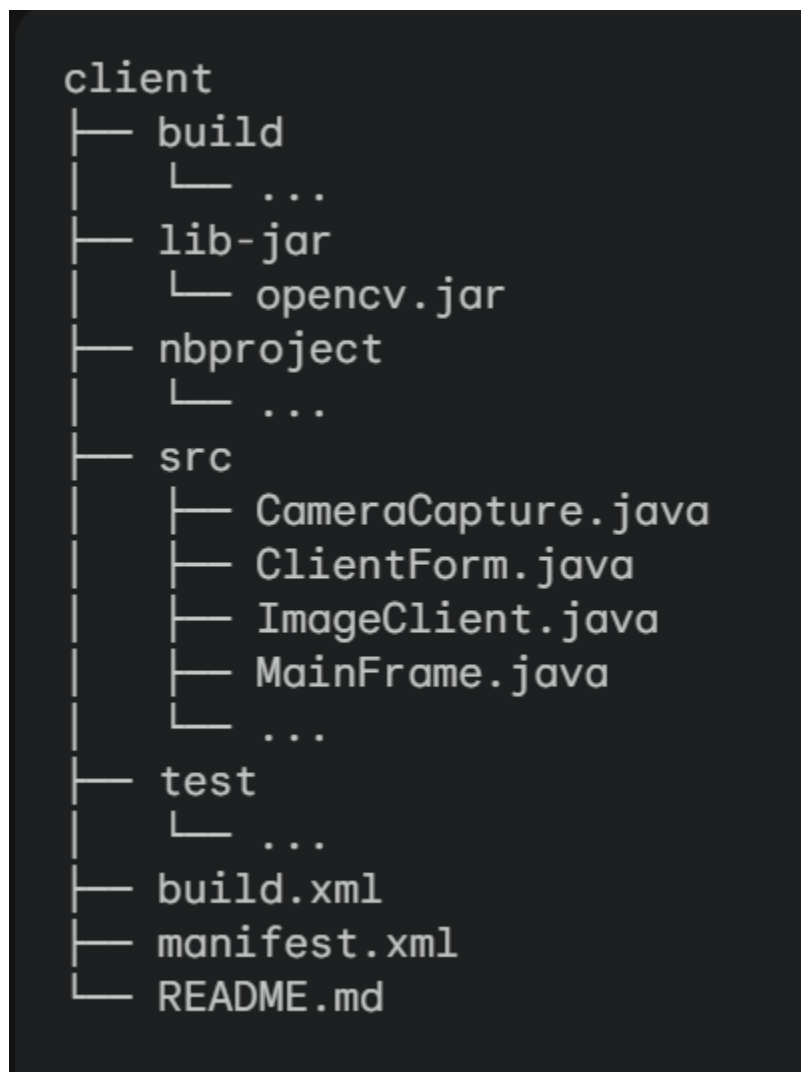
*Bảng 5. Use Case - Ghi dữ liệu vào database*

<b>Name</b>	Ghi dữ liệu vào database
<b>Code</b>	UC05
<b>Short description</b>	UC cho phép server ghi dữ liệu
<b>Actor</b>	Server
<b>Trigger</b>	Actor nhận được thông điệp addUser
<b>Pre-condition</b>	
<b>Post-Condition</b>	Thông báo lưu thành công
<b>Standard process(flow)</b>	1. Actor nhận thông điệp addUser 2. Actor nhận được thông tin người dùng 3. Actor kiểm tra thông tin đã đúng định dạng chưa 4. Actor ghi thông tin vào cơ sở dữ liệu

	<p>5. Actor thông báo cho hệ thống đã lưu thông tin thành công</p> <p>6. Hệ thống hiện thị thông tin đã được lưu</p>
<b>Alternative process</b>	<p>4'. Actor thông báo với hệ thống sai định dạng</p> <p>5'. Hệ thống thông báo ghi sai định dạng thông tin</p>
<b>Error situations</b>	<p>1. Không thể kết nối được với server</p> <p>2. Không kết nối internet.</p>
<b>System state in error situations</b>	

### III. Thiết kế hệ thống

#### 1. Kiến trúc hệ thống



*Hình 2. Cấu trúc client*



```
server
├── build
│   └── ...
├── lib-jar
│   ├── commons-codec-1.14.jar
│   ├── commons-logging-1.2.jar
│   ├── httpclient-4.5.13.jar
│   ├── httpcore-4.4.13.jar
│   ├── httpmime-4.5.13.jar
│   ├── json-20210307.jar
│   ├── mysql-connector-j-9.0.0.jar
│   ├── opencv.jar
│   └── ...
├── nbproject
│   └── ...
├── src
│   └── FaceRecognitionServer.java
├── test
│   └── ...
├── build.xml
├── manifest.xml
└── README.md
```

*Hình 3. Cấu trúc Server*

## 2. Thiết kế chi tiết

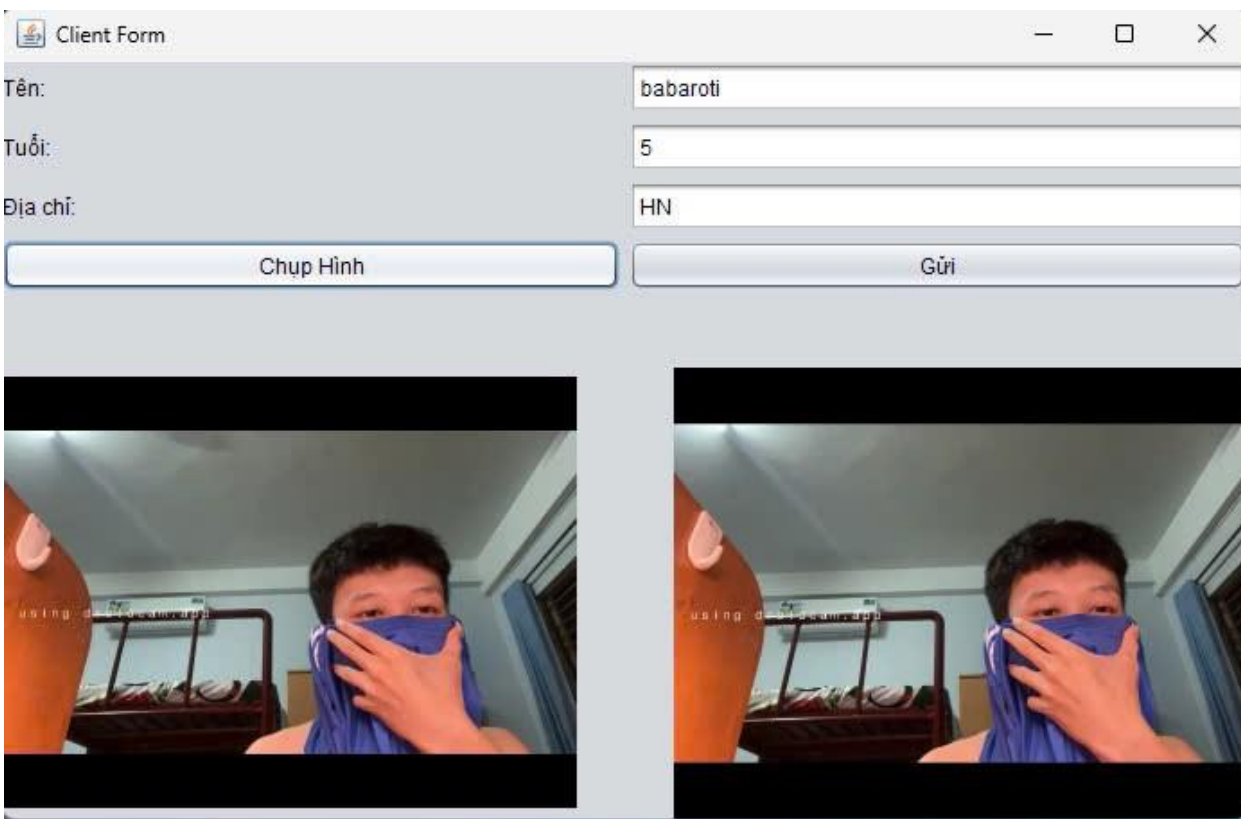
### Thiết kế giao diện người dùng

- Giao diện trang chủ có 3 nút để chuyển đến 3 chức năng khác nhau: Chức năng nhận diện bằng Camera, chức năng nhận diện bằng hình ảnh và chức năng thêm người dùng vào cơ sở dữ liệu.



*Hình 4. Trang chủ client*

- Giao diện trang thêm dữ liệu cho phép người dùng nhập các thông tin như: tên, tuổi, địa chỉ và chụp lại hình ảnh từ client. Sau đó người dùng sẽ bấm nút gửi để lưu lại thông tin dữ liệu từ server



Client Form

Tên: babaroti

Tuổi: 5

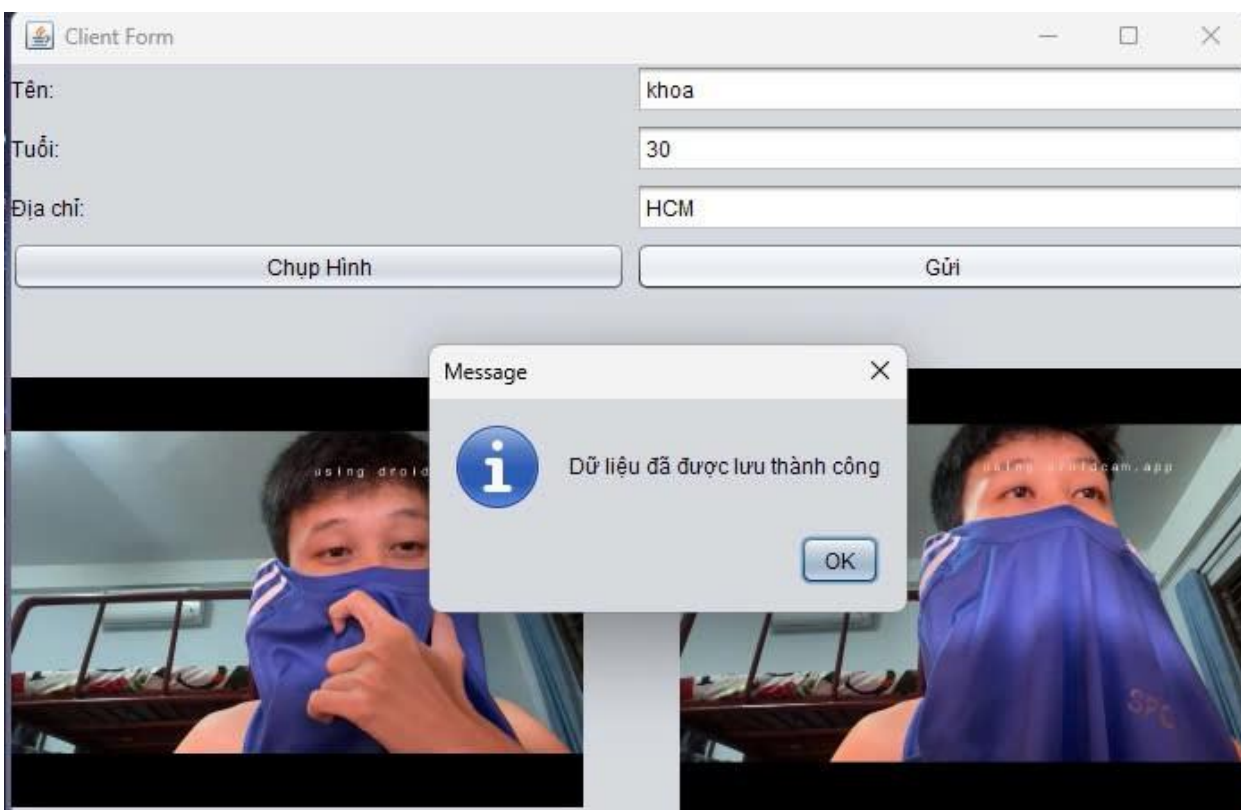
Địa chỉ: HN

Chụp Hình Gửi

using d:\exam\app

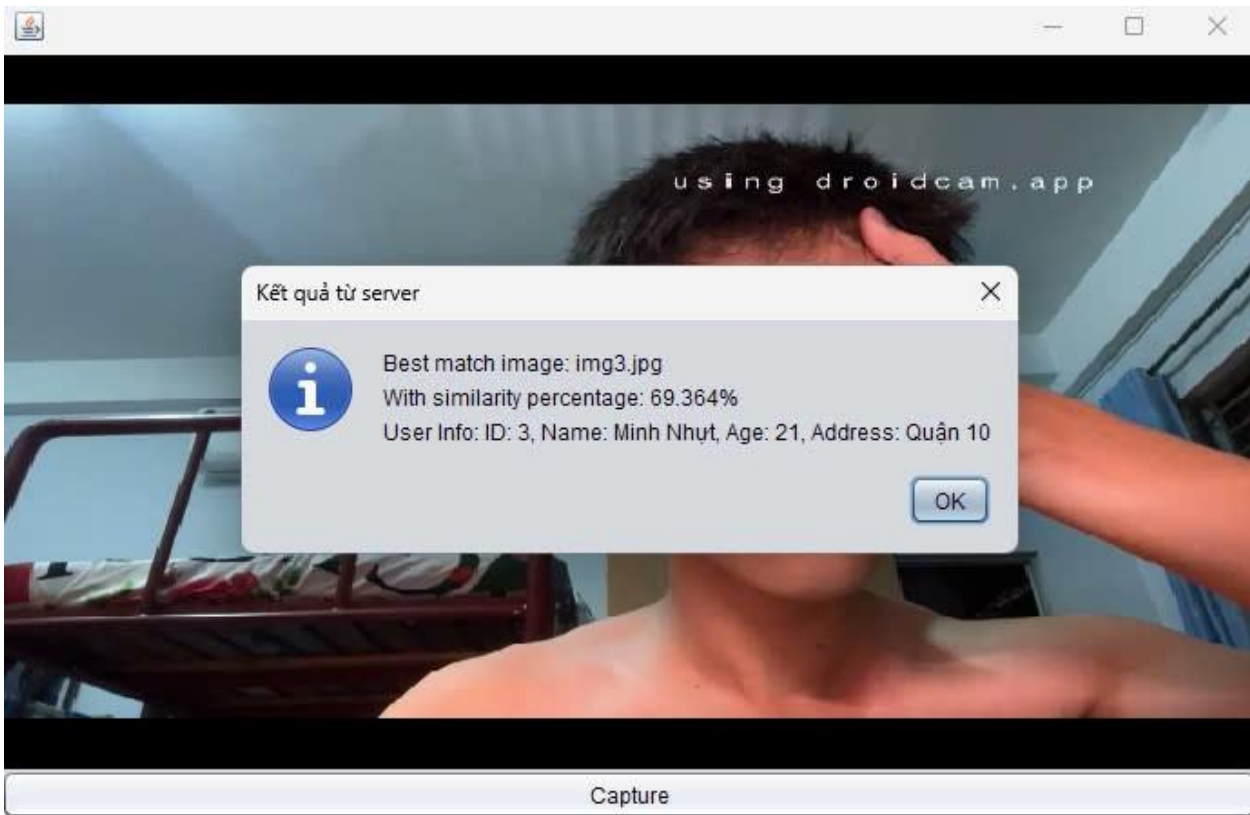
*Hình 5. Trang thêm dữ liệu*

- Giao diện thêm dữ liệu thành công sẽ hiện ra thông báo được trả về từ server và hiện thị lên trên client “Dữ liệu đã lưu thành công”



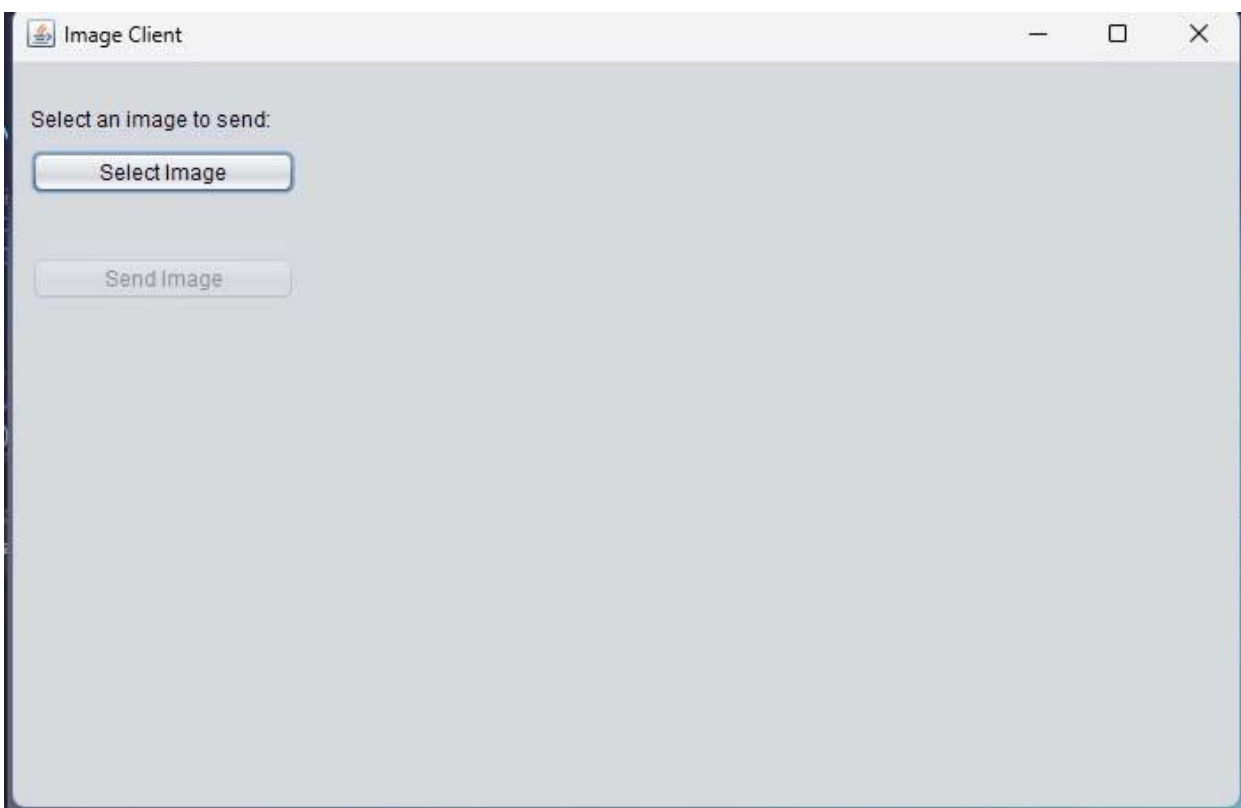
**Hình 6. Trang kết quả sau khi thêm dữ liệu**

- Giao diện nhận diện bằng camera khi bật lên sẽ hiện thị khung quay camera và nút capture khi bấm nút capture client sẽ gửi hình ảnh về server. Server sẽ so sánh và trả về kết quả là thông tin của người dùng có khuôn mặt gần giống nhất.



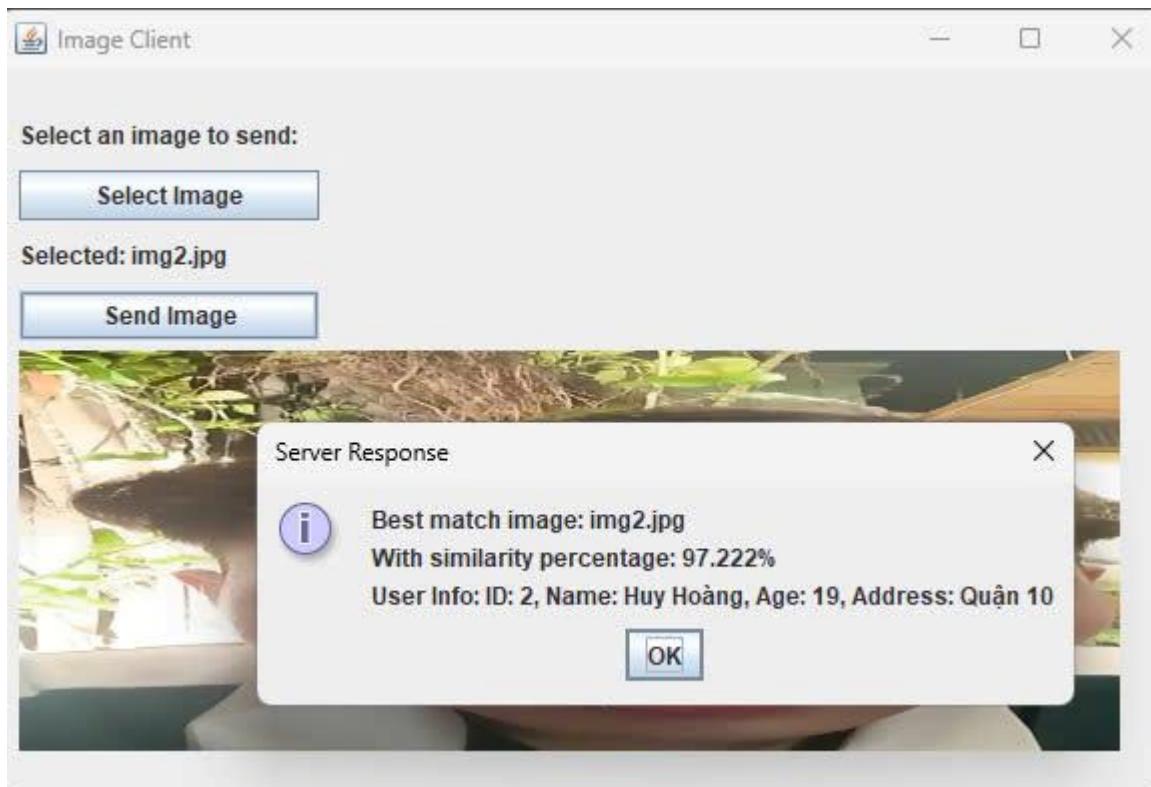
**Hình 7. Trang nhận diện bằng camera và kết quả**

- Giao diện chọn ảnh để kiểm tra kết quả sẽ có nút select image để mở thư mục ra và chọn hình ảnh mình cần nhận diện, sau khi đã chọn xong sẽ hiện hình ảnh lên trên màn hình và nút send image sẽ gửi hình ảnh cần nhận diện về cho server.



***Hình 8. Trang chọn ảnh tĩnh để gửi***

- Giao diện kết quả trả về khi nhận diện từ hình ảnh tĩnh. Sau khi nhận được hình ảnh từ client server sẽ xử lý và trả về thông tin của người dùng có nét gần giống nhất.



*Hình 9. Trang kết quả nhận diện sau khi gửi ảnh tĩnh*

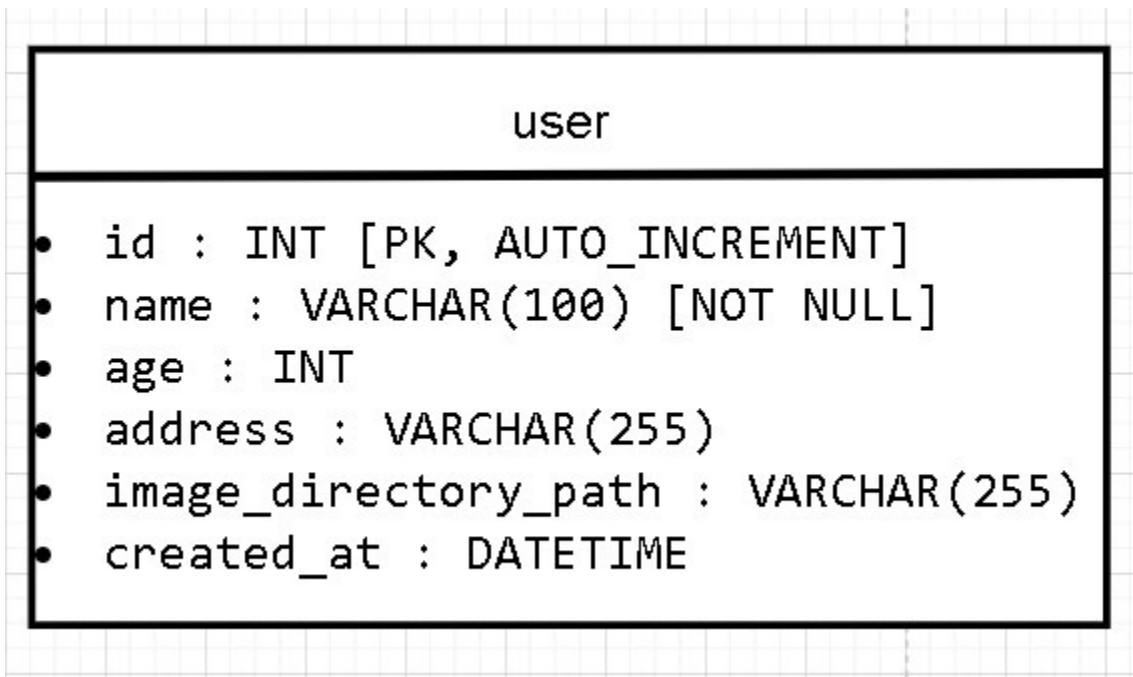
### Thiết kế cơ sở dữ liệu

- Lưu trữ ảnh khuôn mặt của người dùng và các đối tượng cần nhận diện.
- Cột image\_directory\_path sử dụng để truy cứu hình ảnh được lưu trong server sau đó thì sẽ trả về kết quả dựa trên image\_directory\_path để trả về các kết quả thông tin khác

```

1 • use mysql;
2 • CREATE TABLE users (
3     id INT AUTO_INCREMENT PRIMARY KEY,
4     name VARCHAR(100) NOT NULL,
5     age INT,
6     address VARCHAR(255),
7     image_directory_path VARCHAR(255),
8     created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
9 );
    
```

*Hình 10. Tạo cơ sở dữ liệu*



*Hình 11. ERD cơ sở dữ liệu*

### **Thiết kế các module và các luồng dữ liệu**

#### **\* Module chụp hình ảnh (client):**

- Chụp hình ảnh, bao gồm:
  - Chuyển hình ảnh thành định dữ liệu có thể gửi đi.
  - Gửi hình ảnh đến server.
  - Nhận kết quả từ server.
- Sử dụng thư viện xử lý ảnh như OpenCV.

#### **\* Module chọn ảnh từ thư mục (client):**

- Chọn hình ảnh từ thư viện, bao gồm:
  - Chuyển hình ảnh được chọn thành dữ liệu có thể gửi đi.
  - Gửi hình ảnh đến server.
  - Nhận kết quả từ server.

#### **\* Module thêm dữ liệu (client):**

- Xử lý dữ liệu gửi đi, bao gồm:
  - Chuyển đổi dữ liệu thành định dạng hợp lệ.
  - Gửi dữ liệu đến server.
  - Nhận kết quả từ server đã lưu thành công hay không.
- Sử dụng thư viện xử lý ảnh như OpenCV.



**\* Module xử lý ảnh (server):**

- Xử lý ảnh đầu vào, bao gồm:
  - Chuyển đổi ảnh sang định dạng phù hợp.
  - Cắt, lọc nhiễu, điều chỉnh kích thước ảnh.
  - Trích xuất đặc trưng ảnh.
- Sử dụng thư viện xử lý ảnh như OpenCV.

**\* Module nhận diện đối tượng (server):**

- Phân tích ảnh đầu vào để xác định các đối tượng trong ảnh.
- Sử dụng thư viện face++.
- Trả về danh sách các đối tượng được nhận diện cùng với độ tin cậy.

**\* Module thêm dữ liệu (server):**

- Xử lý dữ liệu đầu vào, bao gồm:
  - Chuyển đổi dữ liệu thành định dạng hợp lệ.
  - Lưu dữ liệu vào database.
- Sử dụng thư viện xử lý cơ sở dữ liệu: connectJ để kết nối với cơ sở dữ liệu.

#### **IV. Mã nguồn**

##### **1. Môi trường phát triển**

- Ngôn ngữ lập trình: Java
- Thư viện xử lý ảnh: OpenCV
- Thư viện học máy: Face++
- IDE: Netbean
- Database: MySql Workbench

##### **2. Cấu trúc mã nguồn**

```
client
├── build
│   └── ...
├── lib-jar
│   └── opencv.jar
├── nbproject
│   └── ...
├── src
│   ├── CameraCapture.java
│   ├── ClientForm.java
│   ├── ImageClient.java
│   ├── MainFrame.java
│   └── ...
├── test
│   └── ...
├── build.xml
├── manifest.xml
└── README.md
```

*Hình 12. Cấu trúc client*



*Hình 13. Cấu trúc Server*

### 3. Quá trình cài đặt và triển khai hệ thống

#### B1: Giải nén thư mục



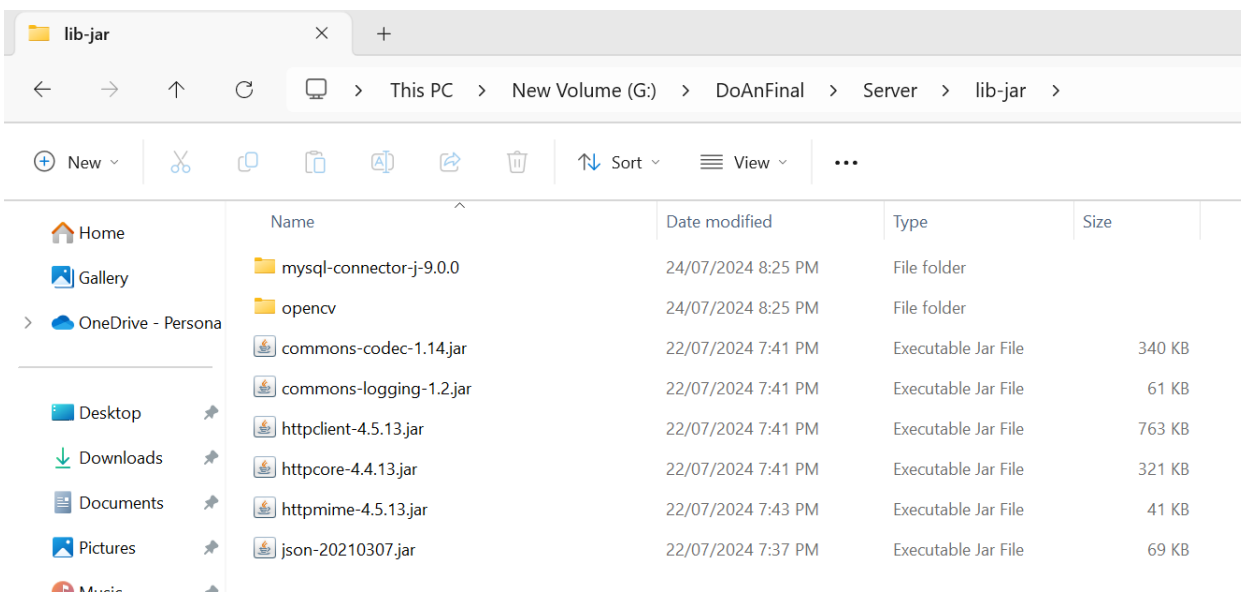
*Hình 14. Thư mục đồ án*

#### B2: Mở netbean và mở project



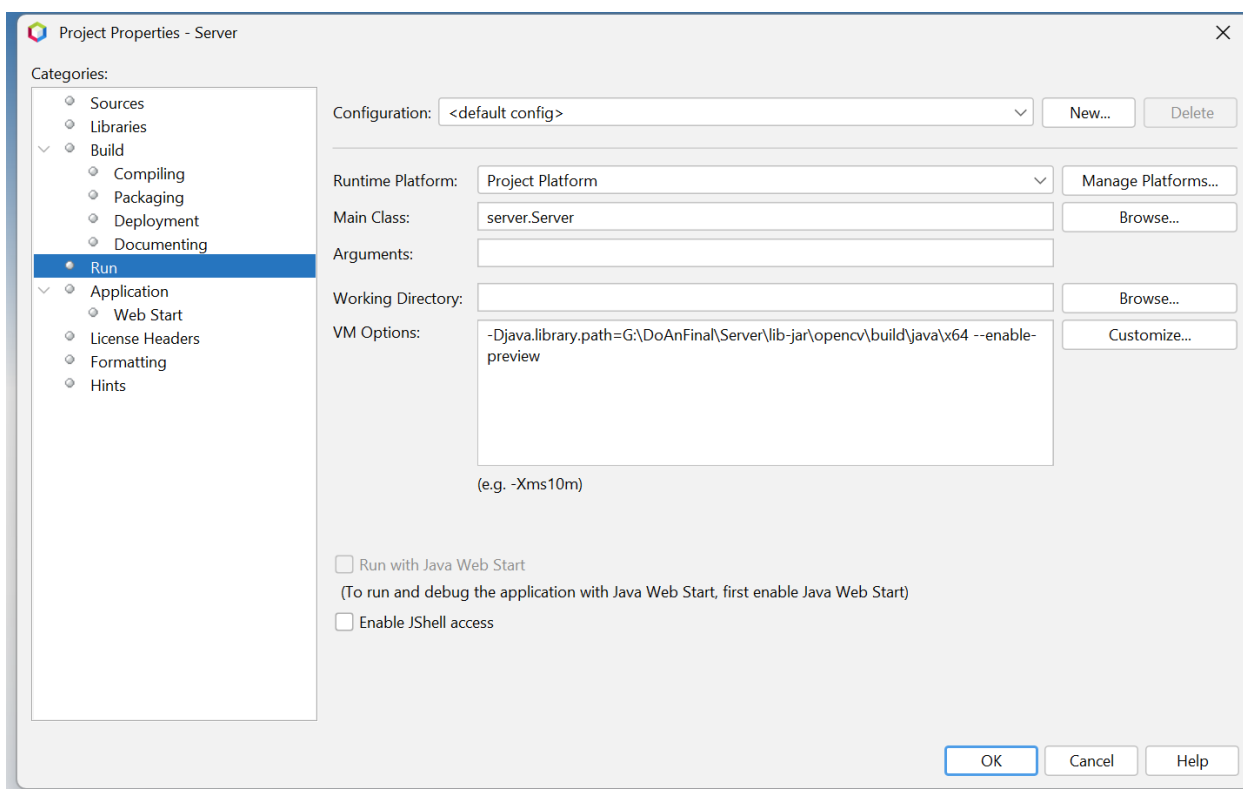
**Hình 15. Mở project lên**

### B3: Add các file jar vào từng project tương ứng



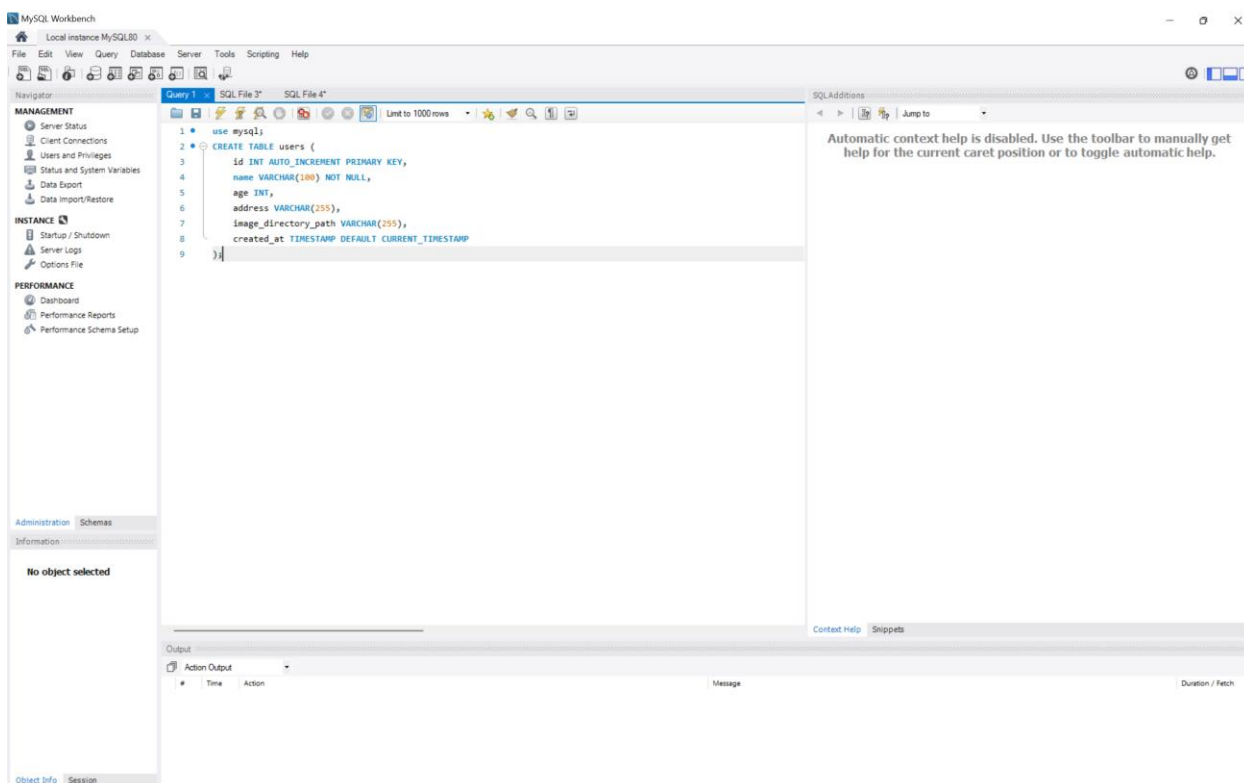
**Hình 16. Cài đặt các file jar này trên mạng về máy**

### B4: Thay đổi đường dẫn tới file opencv\_java4100.dll trong phần VM Options của project



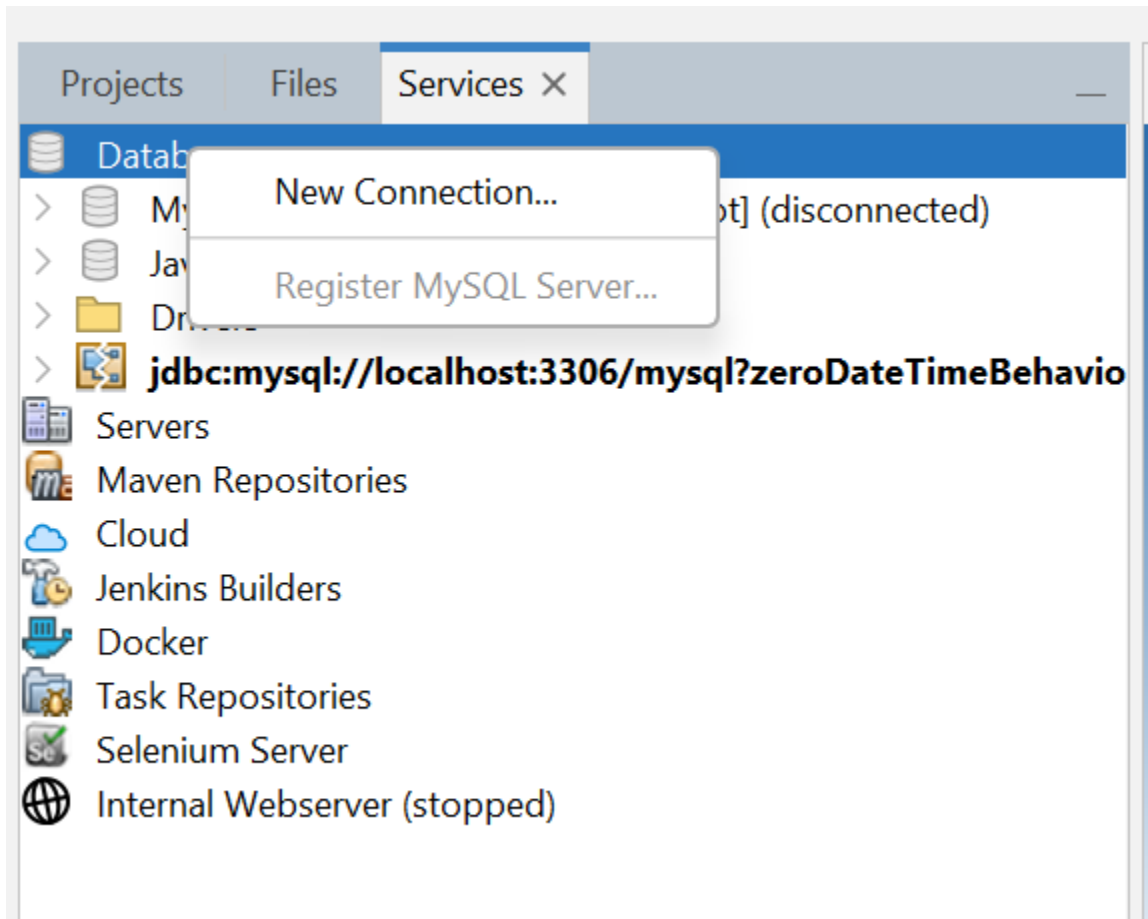
*Hình 17. Thêm thư viện openCV*

## B5: Xây dựng cơ sở dữ liệu

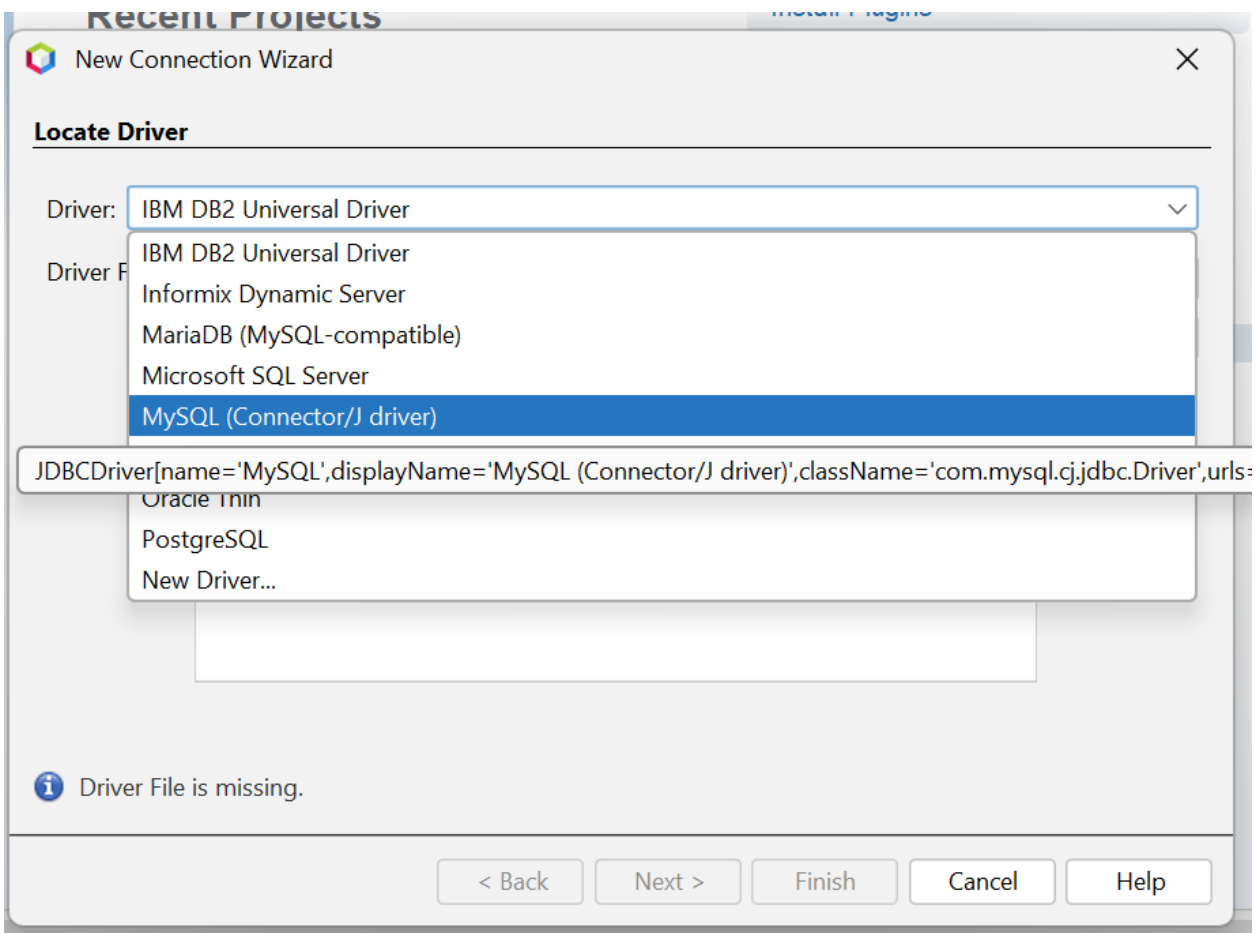


*Hình 18. Tạo cơ sở dữ liệu*

## B6: Kết nối tới cơ sở dữ liệu bằng netbean

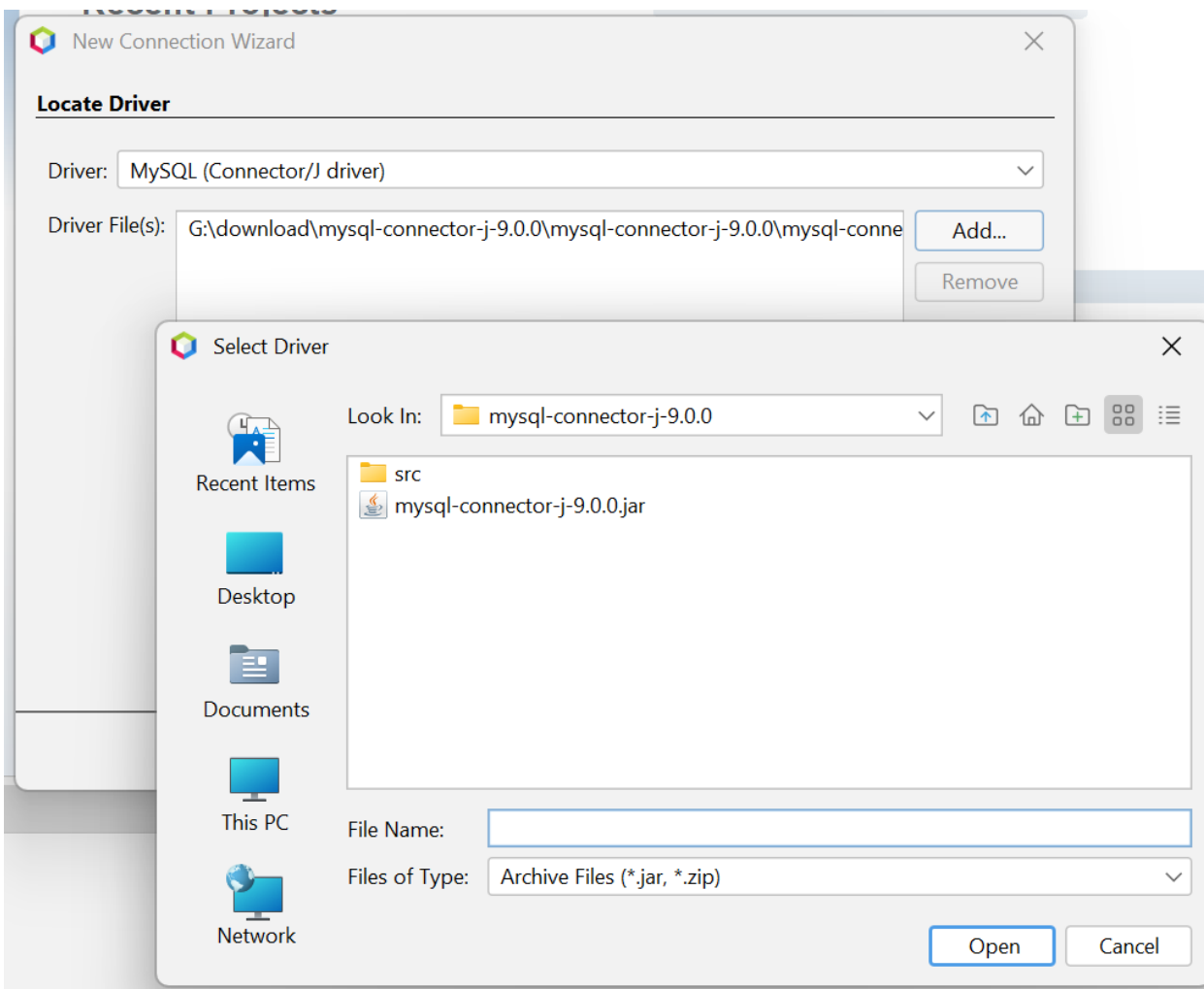


*Hình 19. Chọn kết nối mới*



**Hình 20. Chọn kiểu driver**





**Hình 21. Chọn file jar connect**

**New Connection Wizard**

**Customize Connection**

Driver Name: MySQL (Connector/J driver) ▼

Host: localhost Port: 3306

Database: mysql

User Name: root

Password:

☐ Remember password

Connection Properties Test Connection

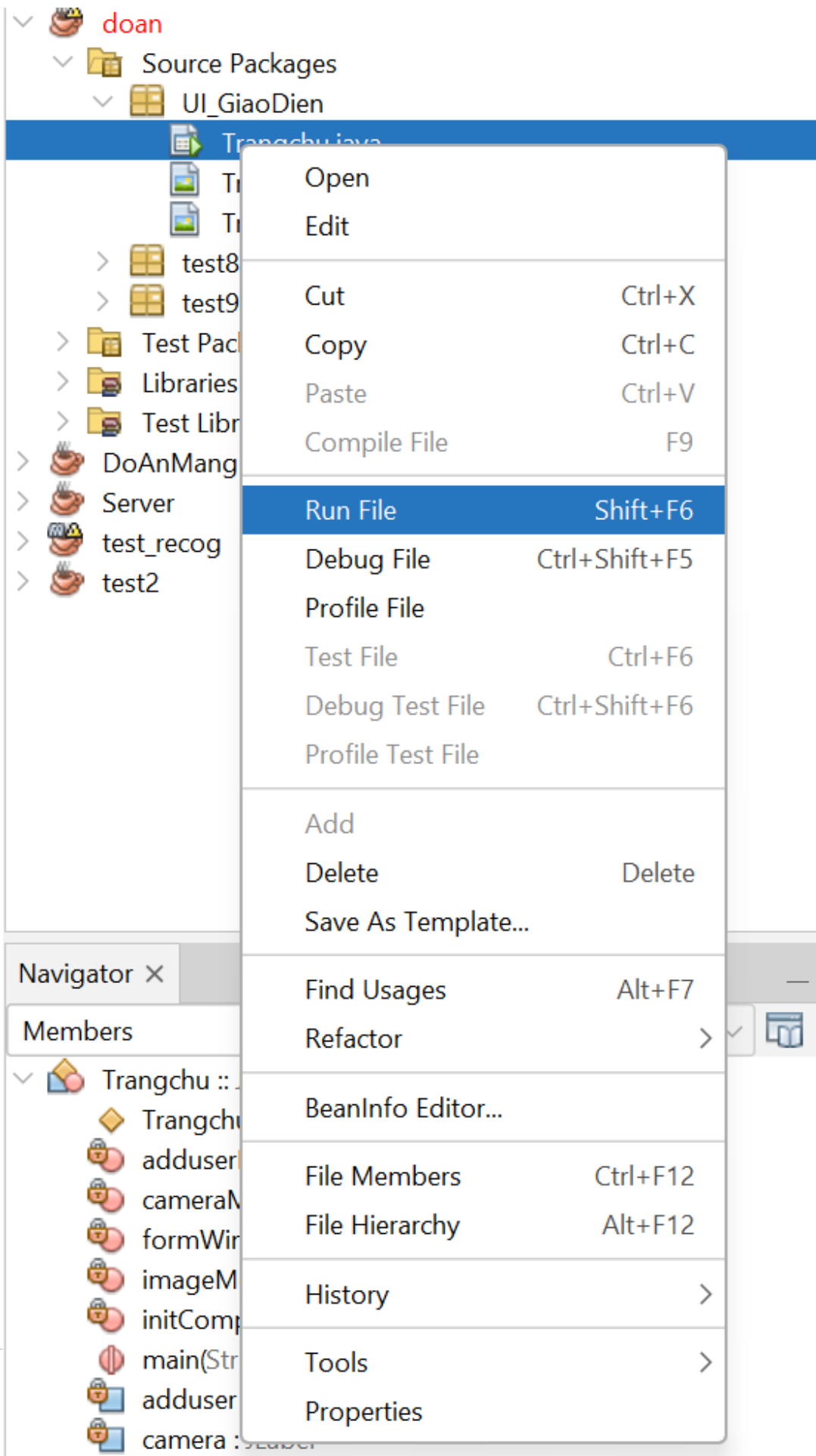
JDBC URL: jdbc:mysql://localhost:3306/mysql?zeroDateTimeBehavior=CONVERT\_TO\_NULL

< Back Next > Finish Cancel Help

*Hình 22. Nhập password và finish*

## Triển Khai hệ thống

### B1: Chọn file giao diện trang chủ

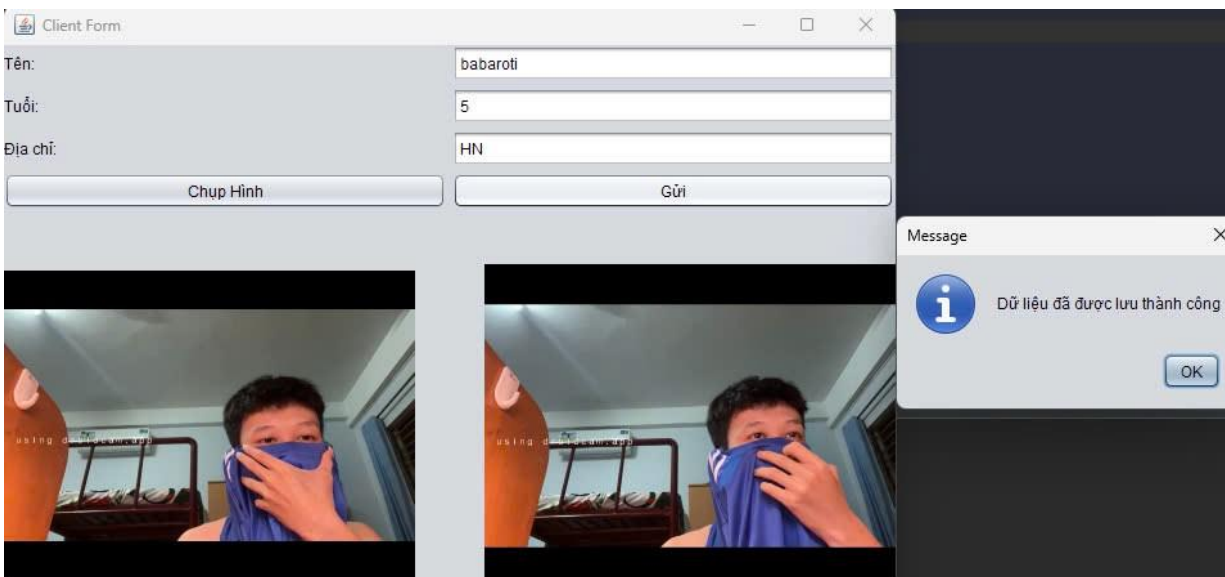


*Hình 23. Chạy giao diện*

**B2: Thêm dữ liệu vào cơ sở dữ liệu bằng cách ấn nút add user**

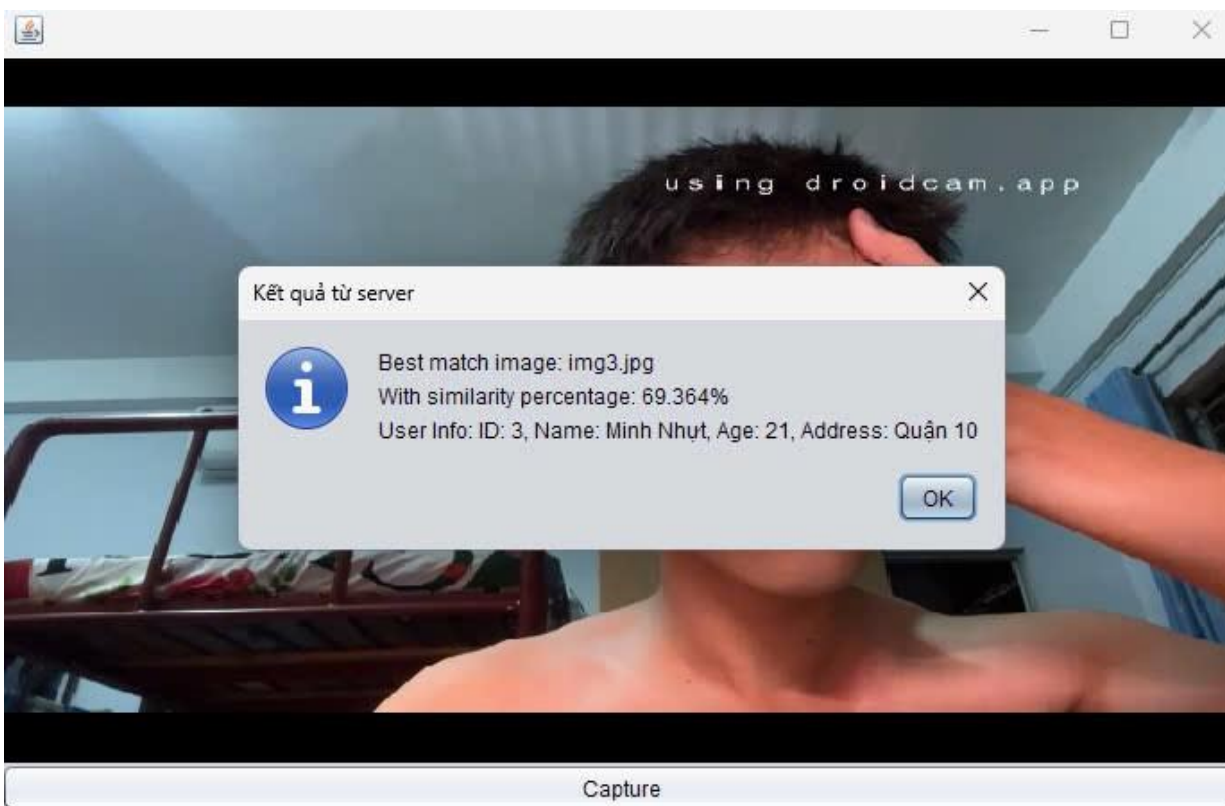


*Hình 24. Trang chủ*



*Hình 25. Nhập thông tin và gửi*

**B3: Khi thêm dữ liệu thành công xong, nhấn nút camera để test tính năng nhận diện khuôn mặt và nhấn nút capture để nhận kết quả nhận diện**



*Hình 26. Nhận diện khuôn mặt camera*

#### 4. Các đoạn mã nguồn quan trọng

```
// Đọc thông điệp từ client
.....
ObjectInputStream ois = new ObjectInputStream(socket.getInputStream());
String message = (String) ois.readObject();
System.out.println("Received message: " + message);

// Xử lý thông điệp
String responseMessage = "";
if (message.equals("addFace")) {
    String ten = (String) ois.readObject();
    String tuoi = (String) ois.readObject();
    String diaChi = (String) ois.readObject();
    byte[] imageBytes = (byte[]) ois.readObject();
    responseMessage = addFace(ten, tuoi, diaChi, imageBytes);
} else if (message.equals("findFace")) {
    byte[] imageBytes = (byte[]) ois.readObject();
    responseMessage = findFace(imageBytes);
} else {
    responseMessage = "Unknown command received";
    System.out.println(responseMessage);
}

// Gửi phản hồi lại client
BufferedWriter writer = new BufferedWriter(new OutputStreamWriter(socket.getOutputStream()));
writer.write(responseMessage);
writer.newLine();
writer.flush();
```

*Hình 27. Đoạn mã nhận dữ liệu và xử lý*

```

private static String saveImage(byte[] imageBytes) throws IOException {
    String directoryPath = "imageData";
    Path directory = Paths.get(directoryPath);
    if (!Files.exists(directory)) {
        Files.createDirectories(directory);
    }

    int imageCount = directory.toFile().list().length + 1;
    String imagePath = directoryPath + File.separator + imageCount + ".jpg";
    try (FileOutputStream fos = new FileOutputStream(imagePath)) {
        fos.write(imageBytes);
    }
    return imagePath;
}

private static void saveToDatabase(String ten, String tuoi, String diaChi, String imagePath) {
    try (Connection conn = DriverManager.getConnection(DB_URL, USER, PASSWORD)) {
        String sql = "INSERT INTO users (name, age, address, image_directory_path) VALUES (?, ?, ?, ?)";
        PreparedStatement pstmt = conn.prepareStatement(sql, Statement.RETURN_GENERATED_KEYS);
        pstmt.setString(1, ten);
        pstmt.setInt(2, Integer.parseInt(tuoi));
        pstmt.setString(3, diaChi);
        pstmt.setString(4, imagePath);

        pstmt.executeUpdate();
        ResultSet rs = pstmt.getGeneratedKeys();
        if (rs.next()) {
            int id = rs.getInt(1);
            System.out.println("Người dùng mới được thêm vào với ID: " + id);
        }
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
}

```

**Hình 28. Lưu hình ảnh và dữ liệu vào cơ sở dữ liệu**

```

private static String compareImageWithDatabase(File image1) {
    File folder = new File("G:\\DoAnFinal\\Server\\imageData");

    File[] files = folder.listFiles();
    double maxConfidence = 0.0;
    String bestMatchImage = "";

    if (files != null) {
        for (File image2 : files) {
            if (image2.isFile() && isImageFile(image2)) {
                try {
                    HttpClient httpClient = HttpClient.createDefault();
                    HttpPost request = new HttpPost(ENDPOINT);

                    FileBody fileBody1 = new FileBody(image1, ContentType.DEFAULT_BINARY);
                    FileBody fileBody2 = new FileBody(image2, ContentType.DEFAULT_BINARY);

                    HttpEntity entity = MultipartEntityBuilder.create()
                        .addTextBody("api_key", API_KEY)
                        .addTextBody("api_secret", API_SECRET)
                        .addPart("image_file1", fileBody1)
                        .addPart("image_file2", fileBody2)
                        .build();

                    request.setEntity(entity);

                    HttpResponse response = httpClient.execute(request);
                    HttpEntity responseEntity = response.getEntity();

                    if (responseEntity != null) {
                        String responseString = EntityUtils.toString(responseEntity);
                        System.out.println("Response from Face++ API: " + responseString);

                        JSONObject jsonResponse = new JSONObject(responseString);

                        if (jsonResponse.has("confidence")) {
                            double confidence = jsonResponse.getDouble("confidence");

                            if (confidence > maxConfidence) {
                                maxConfidence = confidence;
                                bestMatchImage = image2.getName();
                            }
                        } else {
                            System.out.println("Không tìm thấy trường confidence trong phản hồi từ API.");
                        }
                    }
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}

```

Hình 29. Nhận diện khuôn mặt

## 5. Mô tả các điểm nổi bật trong mã nguồn

- Sử dụng các thuật toán học máy tiên tiến để đảm bảo độ chính xác cao cho việc nhận diện khuôn mặt và đối tượng.
- Thiết kế mã nguồn modular, dễ dàng mở rộng và bảo trì.
- Sử dụng giao thức truyền thông an toàn và hiệu quả giữa client và server.

## V. Đánh giá và kết luận

### 1. Kết quả thực hiện

Đã gửi được hình ảnh và chụp được ảnh từ client và gửi về cho server sau khi server xử lý thì sẽ đã trả về được kết quả cho client.

[1]: [\(9\) FINAL - YouTube](#) (kết quả demo)

[2]: [DETAI3\\_22DH110298.rar - Google Drive](#) (project's source code)

## 2. Đánh giá hệ thống

- **Hiệu năng:** Khả năng gửi và trả về kết quả vẫn còn hạn chế, vẫn chưa thể nhận diện với một số lượng lớn người.
- **Bảo mật:** Vẫn chưa có biện pháp bảo mật tốt cho hệ thống.
- **Độ tin cậy:** Hệ thống ổn định, ít sinh ra lỗi khi gửi dữ liệu từ client qua server.

## 3. Kết luận

Đồ án đã thành công trong việc xây dựng một chương trình nhận diện khuôn mặt và đối tượng với đầy đủ các chức năng đề ra. Hệ thống có giao diện người dùng đơn giản, dễ sử dụng, hoạt động hiệu quả, chính xác và đảm bảo an toàn. Kết quả nghiên cứu này có thể áp dụng để phát triển các ứng dụng thực tế trong lĩnh vực an ninh, quản lý, giám sát, v.v.

## 4. Bảng phân công

*Bảng 6. Bảng phân công công việc*

Thành viên	Phân Công	Đánh giá
<b>Bảo</b>	<ul style="list-style-type: none"> <li>Viết báo cáo</li> <li>Viết code nhận diện khuôn mặt</li> <li>Viết code lưu vào database</li> </ul>	<b>100%</b>
<b>Nhật</b>	<ul style="list-style-type: none"> <li>Tạo cơ sở dữ liệu</li> <li>Viết code trả về dữ liệu từ nhận diện</li> <li>Xây dựng kết nối client – server</li> </ul>	<b>100%</b>
<b>Đạt</b>	<ul style="list-style-type: none"> <li>Viết code gửi hình ảnh từ client đến server</li> <li>Viết code chụp ảnh bằng camera</li> <li>Xây dựng kết nối client - server</li> </ul>	<b>100%</b>
<b>Hoàng</b>	<ul style="list-style-type: none"> <li>Viết giao diện</li> <li>Tổng hợp code</li> </ul>	<b>100%</b>



[illegible]

**Hình 30. Project Plan 1**

[illegible]

**Hình 31. Project Plan 2**

## VI. Tài liệu tham khảo

**[1] [Face Recognition with Real Time Database | 2 Hour Course | Computer Vision \(youtube.com\)](#)**

**[2] Code Đạo Dã Ôm - Các hệ thống nhận diện khuôn mặt hoạt động ra sao**  
(youtube.com)

[3] [Java: capture image from webcam with openCV || swing || eclipse || \(youtube.com\)](#)

**[4] Overview (OpenCV 4.10.0 Java documentation)**