

VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



Multidisciplinary Project

Project

Face Recognition Door Locking System

| | | |
|-------------|--------------------------|---------|
| Instructor: | Quản Thành Thơ | |
| Student: | - Đinh Hoàng Anh | 1952553 |
| | - Nguyễn Minh Hùng | 1952737 |
| | - Nguyễn Văn Quốc Chương | 1950004 |
| | - Hoàng Nhật Quang | 1852691 |
| | - Nguyễn Văn Quang | 1952937 |

HO CHI MINH CITY, FEBRUARY 2022



Member list & Workload

| No. | Fullname | Student ID | Problems | Percentage of work |
|-----|------------------------|------------|----------|--------------------|
| 1 | Nguyễn Minh Hùng | 1952737 | ... | 20% |
| 2 | Nguyễn Văn Quang | 1952937 | ... | 20% |
| 3 | Nguyễn Văn Quốc Chương | 1950004 | ... | 20% |
| 4 | Hoàng Nhật Quang | 1852691 | ... | 20% |
| 5 | Đinh Hoàng Anh | 1952553 | ... | 20% |



Contents

| | | |
|----------|--------------------------------------|-----------|
| 1 | Overview | 3 |
| 2 | Devices | 3 |
| 3 | Module | 3 |
| 4 | Requirement | 4 |
| 4.1 | Functional Requirement | 4 |
| 4.2 | Non-Functional Requirement | 4 |
| 5 | Use case Diagram | 5 |
| 5.1 | General | 5 |
| 5.2 | AI | 7 |
| 5.3 | Application | 8 |
| 5.4 | Device | 12 |
| 6 | Mock-up design | 14 |
| 6.1 | Welcome screen | 14 |
| 6.2 | Login/Sign up screen | 14 |
| 6.3 | Main screen | 15 |
| 6.4 | Profile | 16 |
| 6.5 | Member manage | 16 |
| 7 | Database Design | 17 |
| 7.1 | EER Diagram | 17 |
| 7.2 | Relational Diagram | 18 |

1 Overview

Nowadays, private security is one of the most important things that people care about. While traditional doors require key or RFID card to unlock, our project, Smart Door Locking using Face Recognition, will use AI algorithm to unlock. Smart doors are always a comfortable and easy-to-operate choice for security places such as companies, homes, and offices because AI and IoT can be combined for doors. Our team chose this issue because of its practicality as well as the fresh information about AI and IoT that we may gain from it. We know the pros and disadvantages of the problem, as well as how to expand the topic into a larger and better project, in addition to being able to detect problems relating to the topic. In short, this is a very worthwhile project and the results are something to be excited about.

2 Devices

Input devices: Touch button, Temperature sensor, Camera (self-prepared), magnetic switch.

Output devices: Buzzer (speaker for announce), LCD screen for printing status of system.

| Device | ID | Description |
|-----------------|-----------------|---|
| Webcam | INPUT (prepare) | Capture person's face. |
| Buzzer | OUTPUT 2 | Speaker to announce . |
| LCD I2C | OUTPUT 3 | To display door status (locked or unlocked) |
| Touch button | INPUT 5 | Push to capture |
| DHT11 | INPUT 7 | It can sense the room's temperature |
| Magnetic switch | INPUT 8 | It can detect whether the door is locked or not |

Table 1: Devices list

3 Module

The project is divided into 5 modules:

- Module 1: Receiving data from input device: camera (image of users) and magnetic switch (door's status).
- Module 2: Checking the people is trustful or not by AI model.
- Module 3: Sending signal to output device: LCD (text) and Buzzer (announce or not)

- Module 4: Recording access history to Database (Firebase).
- Module 5: Retrieving records via Mobile app.

4 Requirement

4.1 Functional Requirement

Image capture module:

- Capture the image and send it to the smart-phone immediately when user pushes on touch button.
- Buzzer signals sound **once** when user pushes on touch button and the LCD screen will announce "Capturing...".
- Buzzer signals sound **twice** and the LCD screen will announce "Hello [User], Welcome home" when detects familiar persons, **3 times** and the LCD screen will announce "Stranger Detected, please recognize again! " when detects strange person.
- Camera automatically stops user from recognition when detects over 10 times (if no familiar face detected).

Server:

- Store user's information: set of images and other usual information(name, age, sex,...).
- Store User's manual (how to use the system).

Application:

- Data management: Users can manage all of data such as user's profile, recognition history.
- Recognition history: Admin can check the access history of all other users.
- The application can run in background's mode.

4.2 Non-Functional Requirement

- Total response time (from starting recognize to action that open the door or not) is not greater than 5s.
- Application can run on Android ≥ 2.0
- Usability: UI should be easy to use so that users can learn to use in five minutes.

- Security: The system will deactivate automatically if user enter wrong password five times consecutively.
- Recognition accuracy $\geq 90\%$
- Server can support and contain 100 users in the database.

5 Use case Diagram

5.1 General

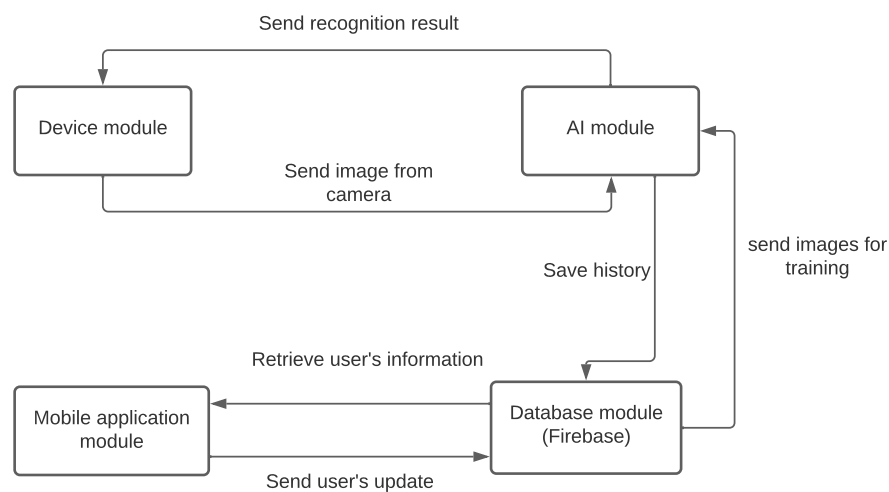


Figure 1: Architecture Diagram

This is use case diagram of the whole system.

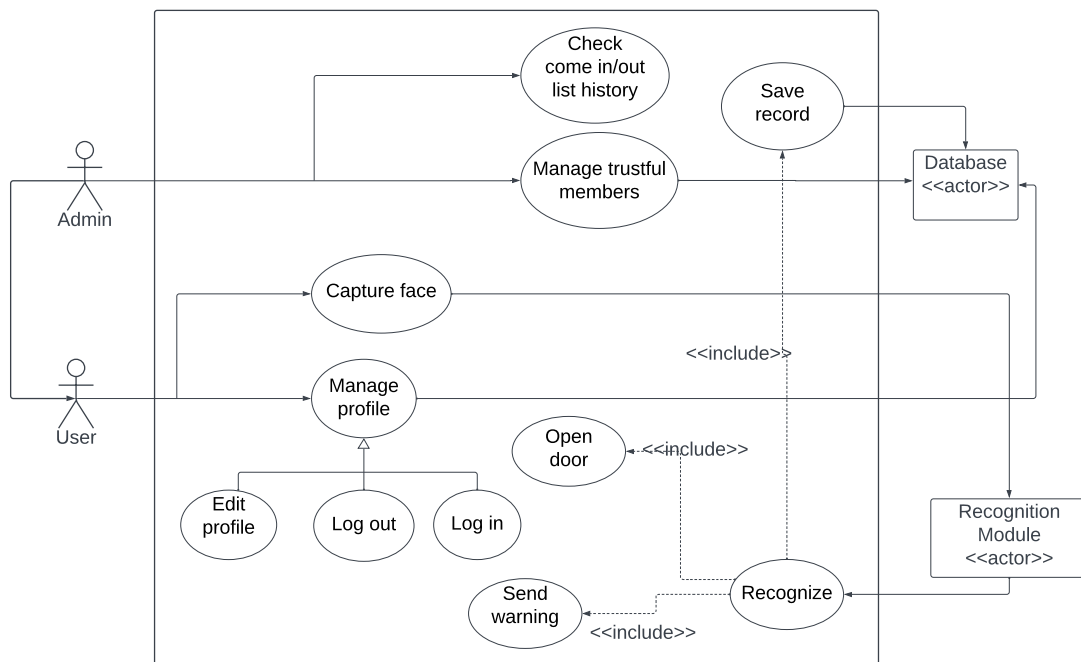


Figure 2: Use case diagram of the whole system

5.2 AI

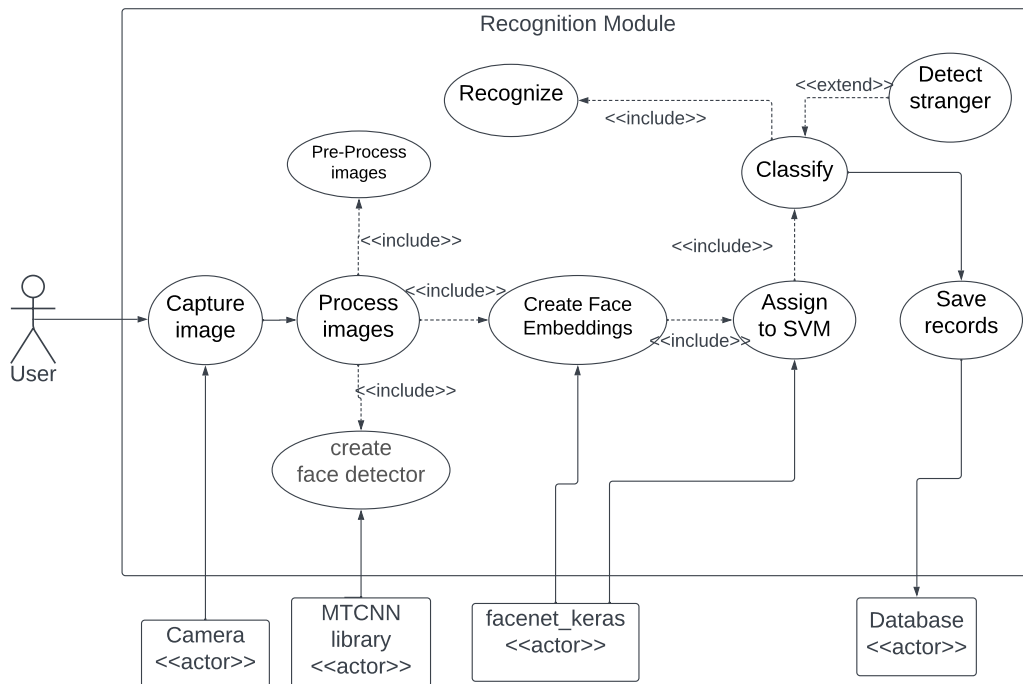


Figure 3: Use case diagram - Face recognition

| | |
|-----------------|---|
| Author | Nguyễn Minh Hùng |
| Use case ID | 0 |
| Use case name | Face Recognition |
| Description | Receive image, process and send the result to output devices |
| Actor(s) | Recognition module |
| Pre-conditions | Camera is active |
| Post-conditions | The module send the recognition's result to controlling module corresponding to images captured |
| Normal flow | <ol style="list-style-type: none"> 1. The camera captured the image and send it to the module. 2. This module perform image pre-processing. 3. The MTCNN library created a face detector and send back to the module. 4. Face_net model creates face embeddings from images and evaluate distance then classify. 5. This module send result to controlling module. |
| Exception flow | <p>Exception 1: at step 3, detect missing face.</p> <p>3a. This module comes back to step 1, user has to capture images again.</p> |

5.3 Application

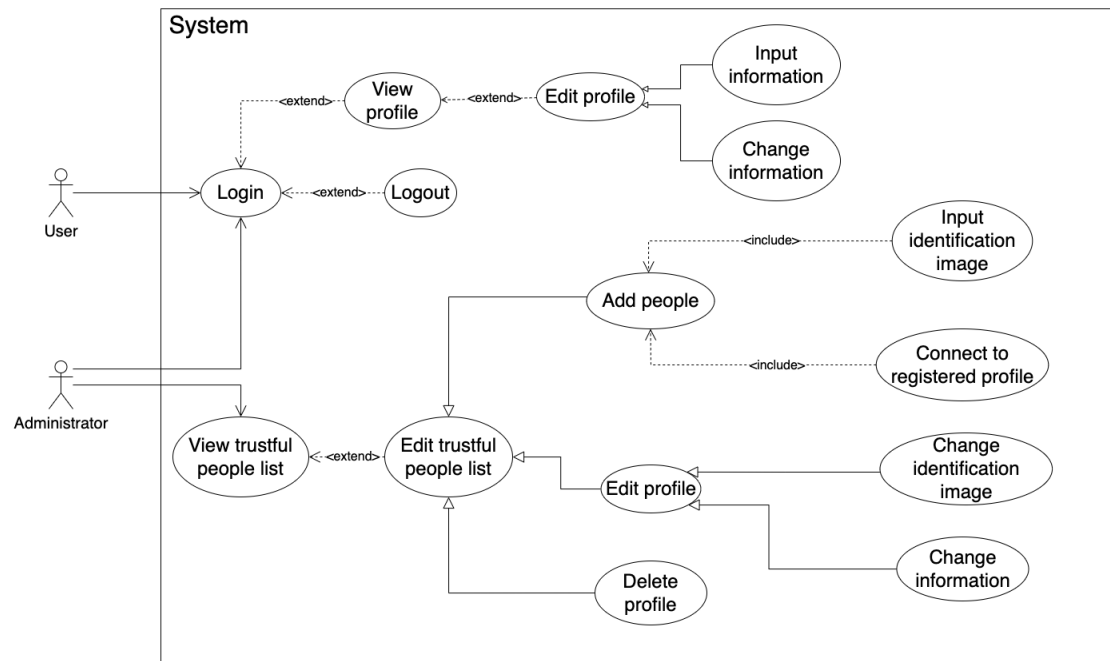


Figure 4: Use case diagram - Edit profile and edit trustful people list

| | |
|-----------------|---|
| Author | Nguyễn Văn Quốc Chương |
| Use case ID | 1 |
| Use case name | Login |
| Description | This use case allows users to login to the application. |
| Actor(s) | User |
| Pre-conditions | None |
| Post-conditions | Admin/User will be logged in to the application |
| Normal flow | 0. Use case begins when admin/user open the application. 1. The system displays a form for the user to input their login credentials <ul style="list-style-type: none"> • Username • Password 2. Users clicks on “Login”. 3. The system will check with if the details are correct. |
| Exception flow | Exception 1: Wrong login credentials at step 3. 3a. This module comes back to step 0, user has to input their information again. |



| | |
|-----------------|---|
| Author | Hoàng Nhật Quang |
| Use case ID | 2 |
| Use case name | Logout |
| Description | This use case allows users to logout to the application. |
| Actor(s) | User |
| Pre-conditions | User must be logged in to the application. |
| Post-conditions | Admin/User will be logged out of the application |
| Normal flow | 0. Use case begins when admin/user clicks on "Logout". 1. The system will move the user session and the user will be logged out. |
| Exception flow | None |

| | |
|------------------|---|
| Author | Nguyễn Minh Hùng |
| Use case ID | 3 |
| Use case name | View profile |
| Include use case | Login |
| Extend use case | View profile |
| Description | This use case allows users to view their information. |
| Actor(s) | User |
| Pre-conditions | User must be logged in to the application. |
| Post-conditions | None |
| Normal flow | 0. Use case begins when user clicks on the "profile" button main screen. 1. The system displays all information of the user. |
| Exception flow | None |

| | |
|------------------|---|
| Author | Dinh Hoàng Anh |
| Use case ID | 4 |
| Use case name | Edit profile |
| Include use case | None |
| Extend use case | None |
| Description | This use case allows users to edit their information. |
| Actor(s) | User |
| Pre-conditions | User must be logged in to the application and viewing their information. |
| Post-conditions | The user's profile will be changed. |
| Normal flow | 0. User click on "Edit" button on profile screen. 1. Users inputs the new information in to the box. 2. Users clicks on "Confirm" button. 3. The system will change the information requested by the user. |
| Alternative flow | Alternative 1: At step 2: Users clicks on "Cancel" button. 2a. Use case ends. |



| | |
|------------------|--|
| Author | Dinh Hoàng Anh |
| Use case ID | 5 |
| Use case name | View trustful people list |
| Include use case | Login |
| Extend use case | Edit trustful people list |
| Description | This use case allows admin to view the list of trustful organization's member. |
| Actor(s) | Administrator |
| Pre-conditions | Administrator must be logged in to the application. |
| Post-conditions | None |
| Normal flow | 0. Administrator click on "Member List" button. 1. The system displays the list of all trustful people. |
| Exception flow | None |

| | |
|------------------|---|
| Author | Hoàng Nhật Quang |
| Use case ID | 6 |
| Use case name | Edit trustful people list |
| Include use case | None |
| Extend use case | None |
| Description | This use case allows administrator to edit the trustful people list. |
| Actor(s) | Administrator |
| Pre-conditions | Admin must be logged in to the application and viewing the list of trustful people. |
| Post-conditions | The trustful people list will be changed. |
| Normal flow | 0. Administrator click on "Add" button on the end of the list. 1. The system displays a form for the admin to input the new information. 2. Administrator inputs the information of the new face. 3. Administrator clicks on "Confirm" button. 4. The system will add the new face profile. |
| Alternative flow | Alternative 1: At step 0: Users clicks on "Delete" button on the right of each profile. 1a. The system displays a warning box. 1b. Administrator clicks on "Confirm" button. 1c. The system deletes the member profile that requested by the admin. |



| | |
|-----------------|--|
| Author | Nguyễn Văn Quang |
| Use case ID | 8 |
| Use case name | Check access history |
| Extend use case | Filter name and date |
| Description | Allow users to check access history |
| Actor(s) | Users |
| Pre-conditions | Users must be logged in the application. |
| Post-conditions | Users can see the access history or |
| Normal flow | 0. Users choose the history button. 1. System display history on screen 2. User type name or time. 3. Users choose search. 4. The system will display the access history base on time and name |
| Exception flow | Exception 1: at step 3. system can not find the name in access history 3a. This module comes back to step 2. user has to type the name or leave the box blank. |

5.4 Device

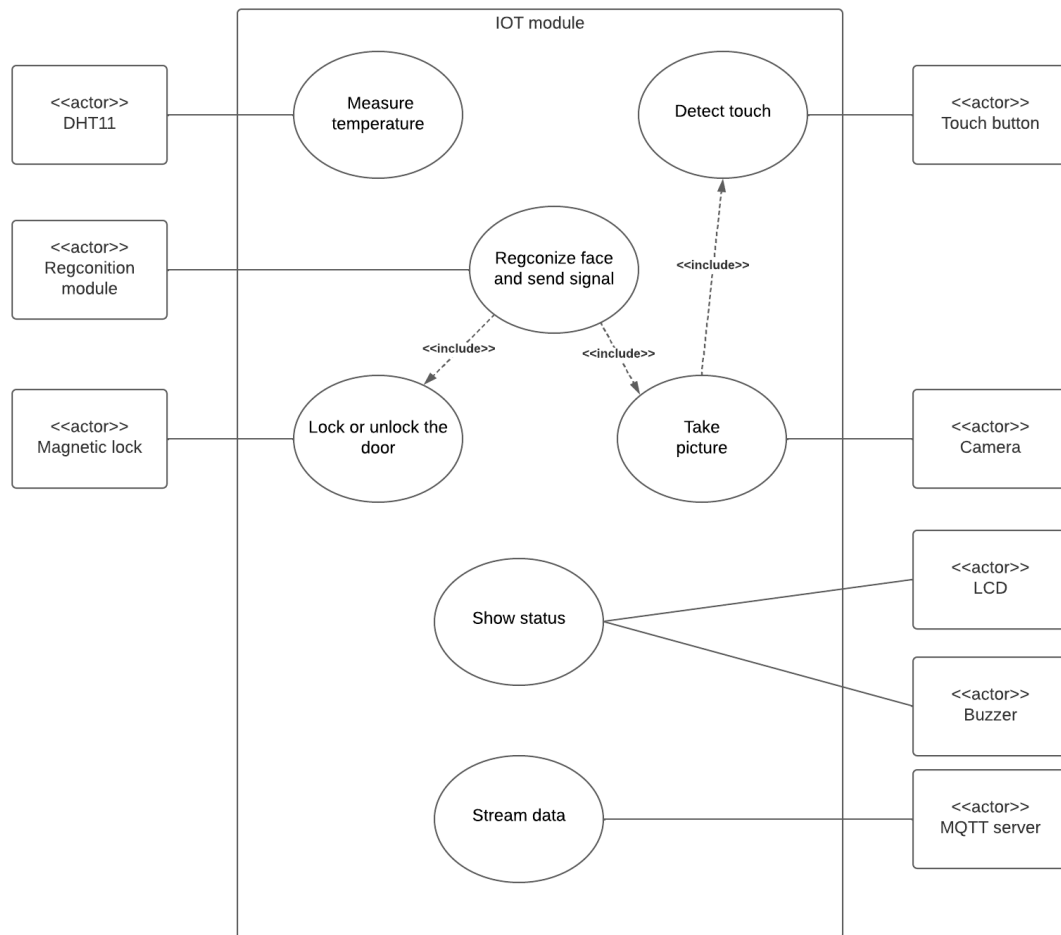


Figure 5: Use case diagram - IOT module

| | |
|-----------------|--|
| Author | Nguyễn Văn Quang |
| Use case ID | 9 |
| Use case name | Measure temperature |
| Description | This use case describes how temperature is measured. |
| Actor(s) | DHT11 |
| Pre-conditions | DHT11 is working properly and connected to Microbit. |
| Post-conditions | Room temperature is measured and saved to database. |
| Normal flow | 1. DHT11 uses the sensor to measure the temperature. 2. Microbit send the data through gateway to MQTT server. 3. Data is saved to database. |
| Exception flow | None |



| | |
|-----------------|---|
| Author | Nguyễn Minh Hùng |
| Use case ID | 10 |
| Use case name | Detect touch |
| Description | This use case describes how touch button is detect. |
| Actor(s) | Touch button. |
| Pre-conditions | Touch button is working properly and connected to Microbit. |
| Post-conditions | Be able to detect the button is touched and send signal to database. |
| Normal flow | 1. Touch button receive data. 2. Microbit send the data through gateway to MQTT server. 3. Data is saved to database. |
| Exception flow | None |

| | |
|----------------------|---|
| Author | Hoàng Nhật Quang |
| Use case ID | 11 |
| Use case name | Take photo |
| Description | This use case describes how temperature is measured. |
| Actor(s) | DHT11 |
| Included use case(s) | Touch button |
| Pre-conditions | The ESP32 is working properly and connected to microbit. |
| Post-conditions | The photo of user is taken and saved to database. |
| Normal flow | 1. The touch button receive signal. 2. Base on the signal of touch button, the camera take photo of the user. 3. Data is saved to database. |
| Exception flow | None |

6 Mock-up design

6.1 Welcome screen

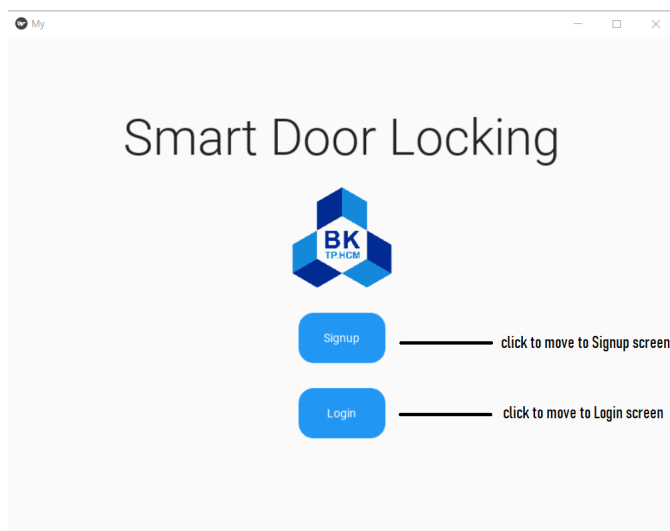


Figure 6: Welcome screen

6.2 Login/Sign up screen

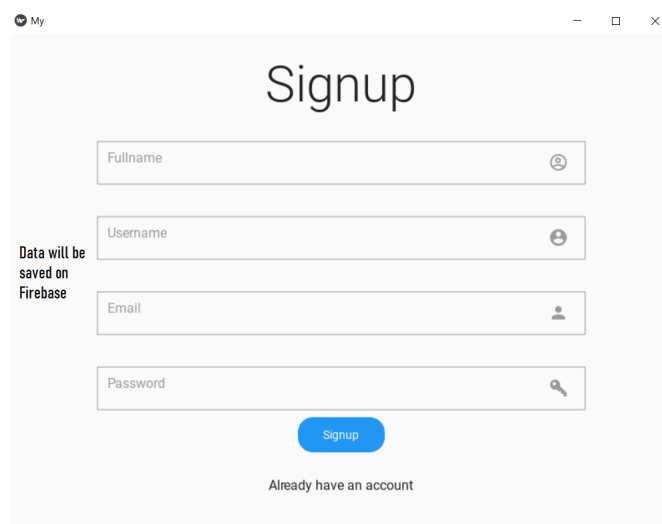


Figure 7: Sign up screen

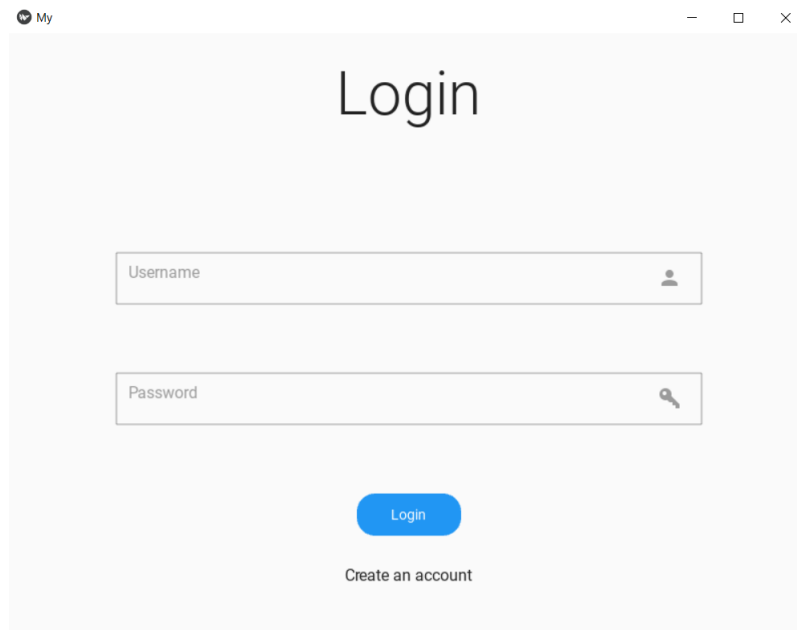


Figure 8: Login screen

6.3 Main screen

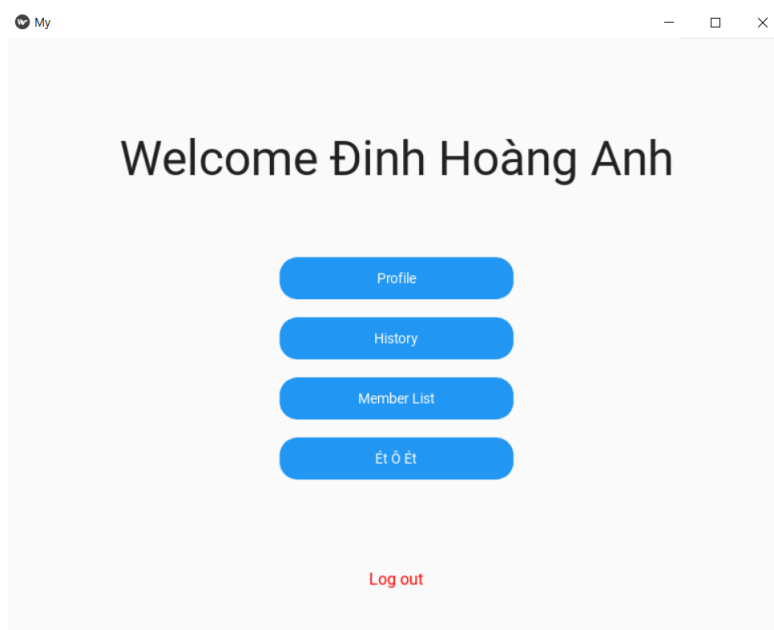


Figure 9: Main screen

6.4 Profile

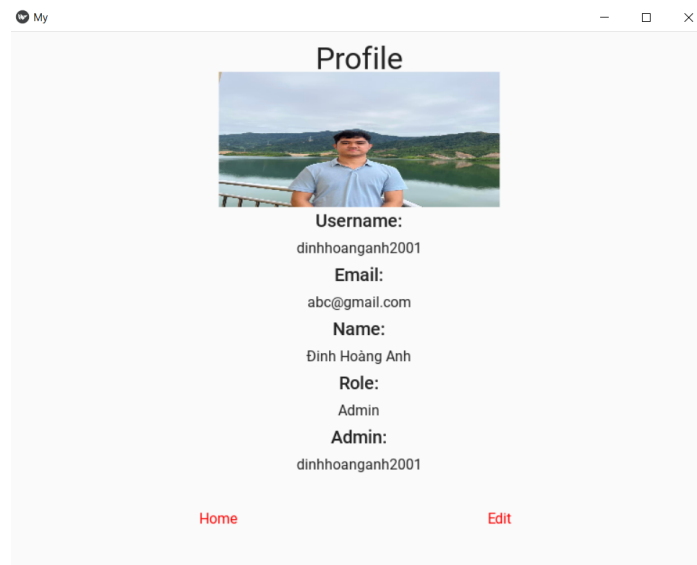


Figure 10: Profile screen

6.5 Member manage

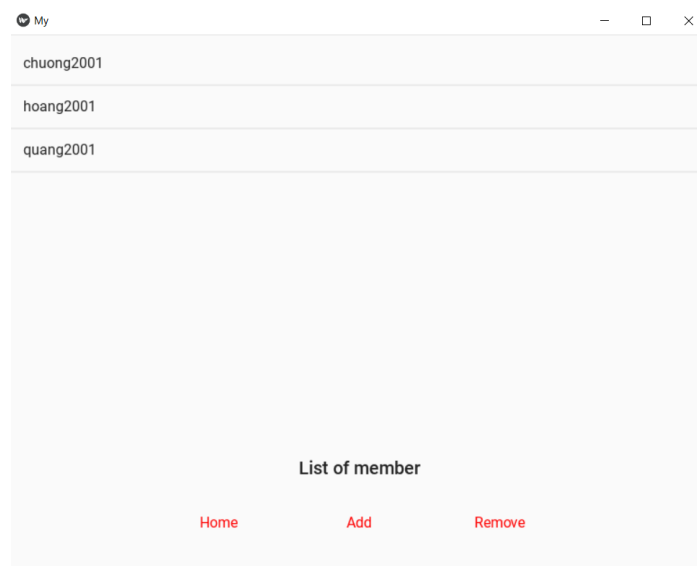


Figure 11: Member manage screen

7 Database Design

7.1 EER Diagram

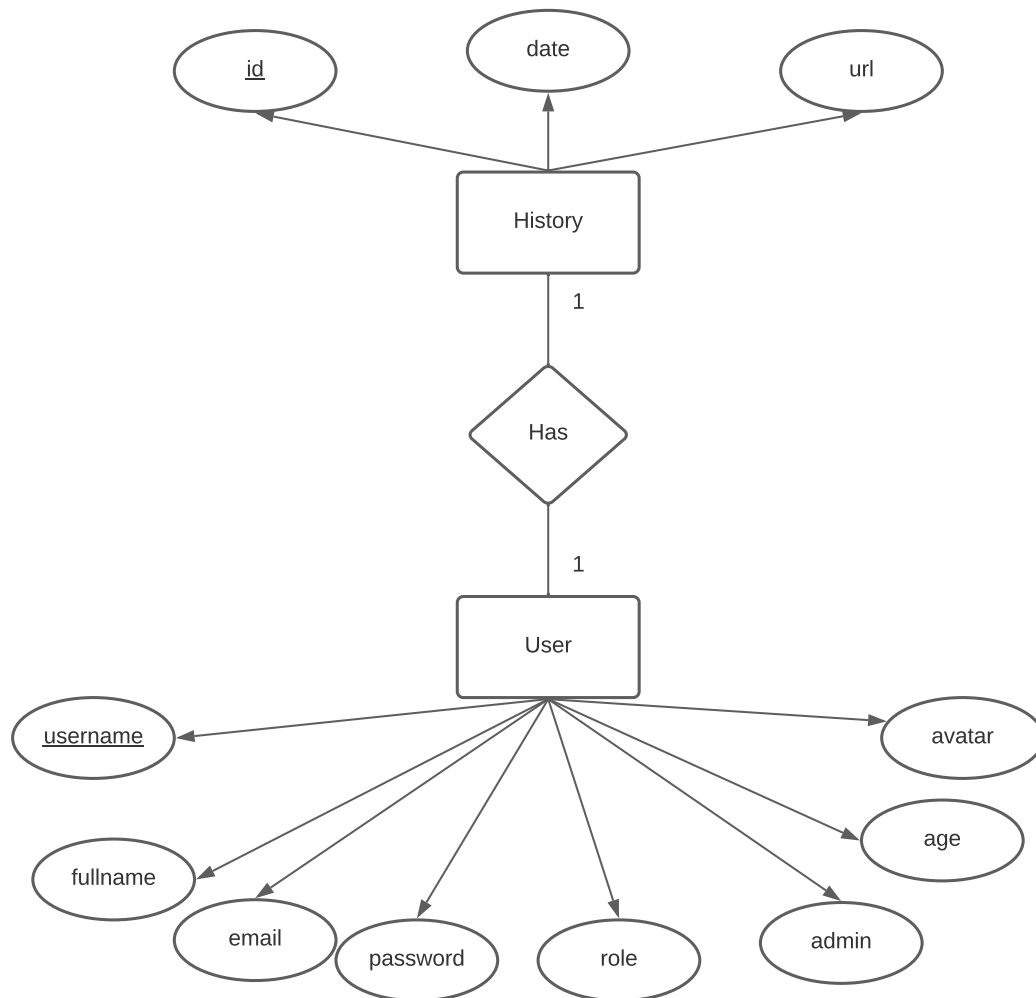


Figure 12: ER diagram for database design



7.2 Relational Diagram

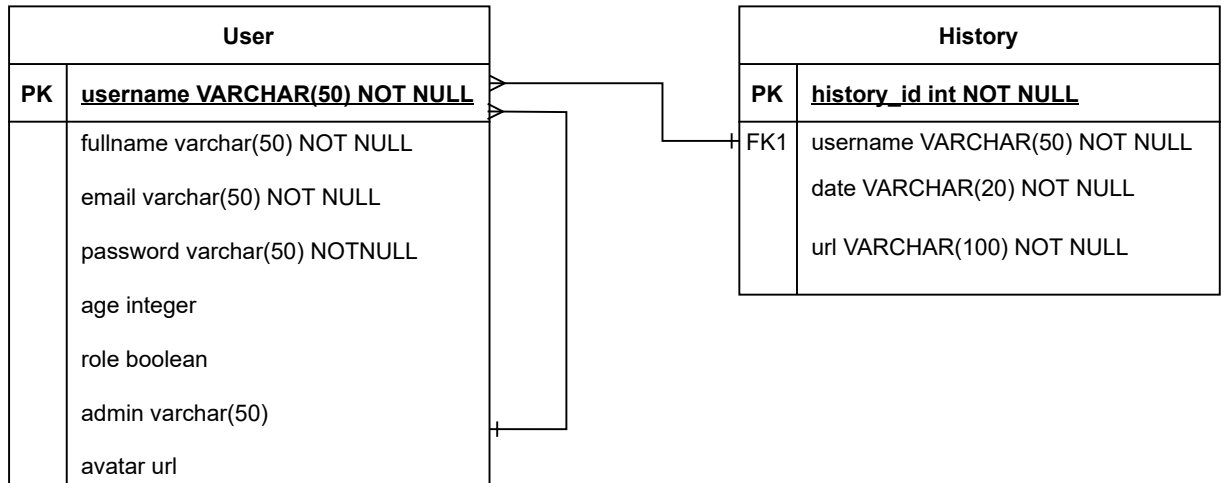


Figure 13: Database schema