# BÁO CÁO ĐỒ ÁN SOCKET WEATHER FORECAST

## Thông tin nhóm:

* Nguyễn Đình Trí – MSSV: 19120405
* Nguyễn Phan Đức anh – MSSV: 19120449

## Đánh giá mức độ hoàn thành:

|  |  |  |
| --- | --- | --- |
| **Chức năng** | **Ý nghĩa** | **Mở rộng** |
| **KẾT NỐI** | Cho phép client kết nối đến server thông qua kết nối TCP | 1. Cho phép n client kết nối đồng thời đến server (n do quản trị server quyết định)  2. Cho phép client và server đặt tại các host khác nhau |
| ***Hoàn thành*** | Checkmark with solid fill | Checkmark with solid fill |
| **QUẢN LÝ KẾT NỐI** | Khi client hoặc server mất kết nối đột ngột, không làm chương trình treo hay xảy ra lỗi | 1. Các client thao tác tuần tự  2. Quản lý kết nối đa tiểu trình |
| ***Hoàn thành*** | Checkmark with solid fill | Checkmark with solid fill |
| **ĐĂNG NHẬP** | 1. Client đăng nhập bằng cách gửi username, password cho server  2. Server nhận thông tin username, password từ client và kiểm tra với thông tin đã lưu trữ tại server |  |
| ***Hoàn thành*** | Checkmark with solid fill |  |
| **ĐĂNG KÝ** | 1. Client đăng ký bằng cách gửi username, password cho server  2. Server nhận thông tin username, password từ client và kiểm tra với thông tin đã lưu trữ tại server, nếu đã tồn tại, gửi thông báo đến client, yêu cầu đăng ký tài khoản khác |  |
| ***Hoàn thành*** | Checkmark with solid fill |  |
| **XEM THỜI TIẾT CÁC THÀNH PHỐ** | 1. Client gửi yêu cầu đến server để nhận về toàn bộ danh sách các thành phố.  2. Server tiếp nhận yêu cầu, gửi về danh sách các thành phố kèm thông tin thời tiết, client trình bày danh sách. | Trả về thông tin thời tiết theo thời gian thực, tuỳ theo ngày giờ truy vấn, tức thông tin truy vấn từ client bao gồm ngày truy vấn. |
| ***Hoàn thành*** | Checkmark with solid fill | Checkmark with solid fill |
| **XEM THÔNG TIN THỜI TIẾT 1 THÀNH PHỐ** | Client gửi ID của thành phố lên server, server tiếp nhận và trả về thông tin chi tiết của thành phố đó. Client hiển thị các thông tin  Thông tin thời tiết cần trả về gồm 7 ngày gần nhất cho thành phố được truy vấn. | 1. Phía server xây dựng thêm một ứng dụng client dành cho admin, cho phép admin cập nhật thông tin thời tiết.  2. Kiểm tra đăng nhập là admin mới được sử dụng tiếp các chức năng của admin.  a) Thêm thành phố mới  b) Cập nhật thông tin thời tiết theo thời gian (ngày)  c) Cập nhật các thông tin thời tiết chi tiết của thành phố theo 7 ngày gần nhất. |
| ***Hoàn thành*** | Checkmark with solid fill | Checkmark with solid fill |
| **QUẢN LÝ CƠ SỞ DỮ LIỆU** | Sử dụng dữ liệu lưu trữ tĩnh trong code  Sử dụng dữ liệu lưu trữ trong file .txt | Sử dụng cơ sở dữ liệu có cấu trúc: xml, json, sql, ... |
| ***Hoàn thành*** | Checkmark with solid fill | Checkmark with solid fill |
| **THOÁT** | 1. Client được phép gửi thông báo ngừng kết nối đến server  2. Server có thể gửi thông báo ngừng kết nối đến tất cả client đang hoạt động |  |
| ***Hoàn thành*** | Checkmark with solid fill | Checkmark with solid fill |
| **GIAO DIỆN** | Console Application | Có thiết kế giao diện đồ hoạ cho chương trình (GUI)   * Client * Server |
| ***Hoàn thành*** | Checkmark with solid fill | Checkmark with solid fill |
| ***Tổng*** | 100% | |

## Kịch bản giao tiếp của chương trình:

1. Giao thức tầng application của chương trình (sau khi đã kết nối TCP):

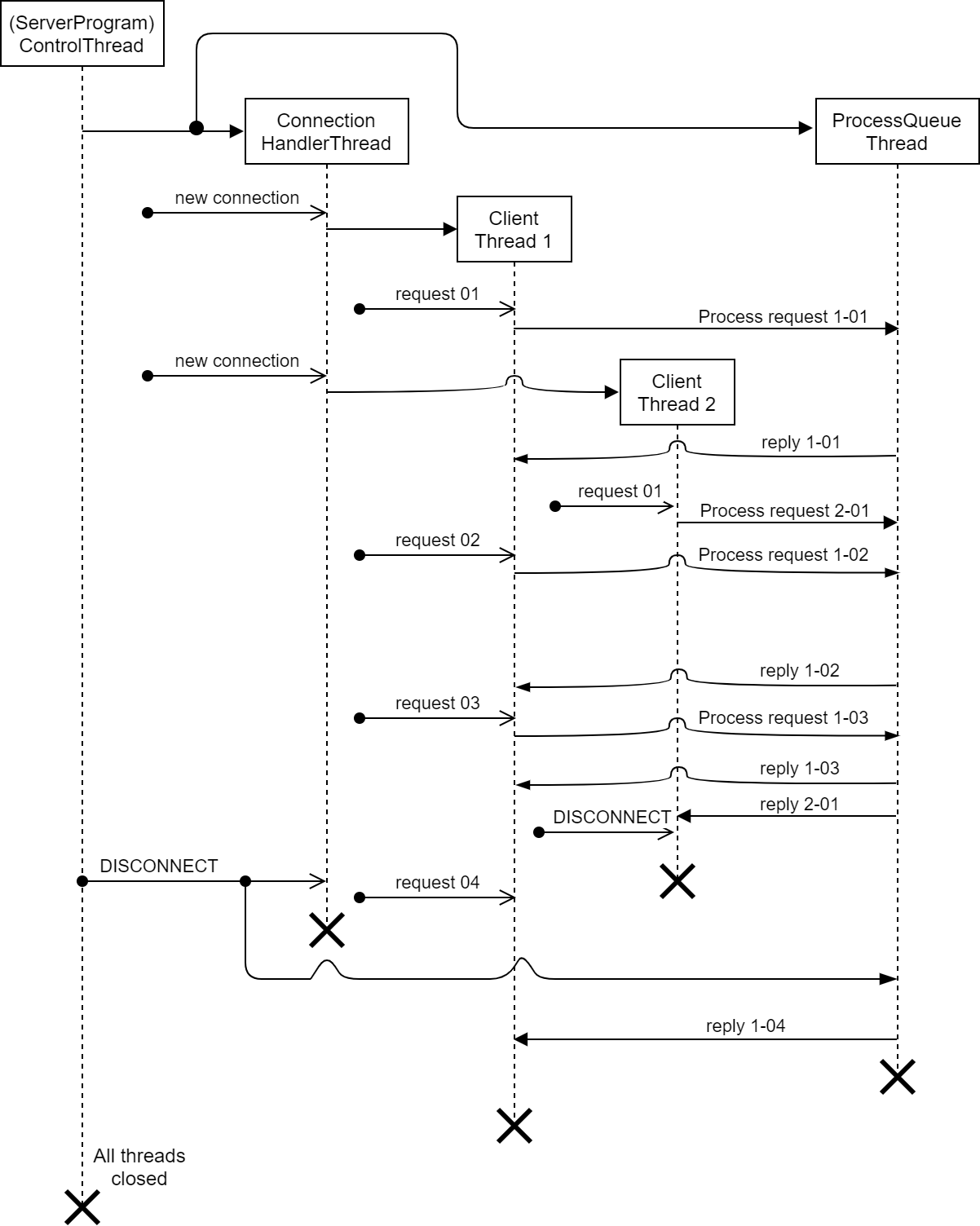
*PHÍA CLIENT*

* Sau khi server mở kết nối, các client có thể kết nối đến host và port của server.
* Sau khi kết nối TCP được hình thành, client có thể:
  + Gửi yêu cầu DISCONNECT để đóng kết nối.
  + Gửi yêu cầu LOGIN để đăng nhập.
  + Gửi yêu cầu REGISTER để đăng ký.
* Nếu client đã đăng nhập thành công, client có thể
  + Gửi yêu cầu DISCONNECT để đóng kết nối.
  + Gửi các yêu cầu WEATHER để truy vấn thông tin từ server
* Trong quá trình client chạy, server có thể yêu cầu hủy kết nối bằng thông điệp DISCONNECT. Server bảo đảm rằng các yêu cầu từ client đã được gửi phản hồi đầy đủ và client có thể gửi lại CONFIRM DISCONNECTION sau khi đã nhận đủ phản hồi mình cần.

*PHÍA SERVER*

*Lưu ý: Một số đối tượng trong source code được đặt tên khác*

* Phần mềm Server khi khởi động có một luồng chính, tạm gọi là ControlThread.
* Khi mở socket server thì tự động tạo ra hai luồng:
  + ConnectionHandlerThread sử dụng
  + ConnectionHandlerThread sẽ lắng nghe các kết nối đến.
  + ProcessQueueThread sẽ liên tục truy cập vào hàng chờ yêu cầu chung của các client để xử lý và gửi lại các câu trả lời.
* Tại ConnectionHandlerThread, mỗi khi có một kết nối mới đến, một luồng mới được tạo ra để lắng nghe các yêu cầu của client đó.
  + Các yêu cầu được gửi chung vào một hàng chờ của ServerProgram.
* Tại ProcessQueueThread, các yêu cầu trong hàng chờ được xử lý tuần tự, lần lượt. Tại đây cũng xảy ra quá trình gửi trả lại thông điệp cho client.
  + Yêu cầu trong hàng đợi được bổ sung thêm một con số định danh cho client, điều này giúp ServerProgram xác định được mỗi yêu cầu là của ai.
* Khi client A gửi một yêu cầu DISCONNECT, ProcessQueueThread sẽ chờ ClientThread đóng rồi mới tiếp tục. Điều này thường không cản trở đến quá trình xử lý, vì Client Thread trong quá trình lằng nghe thông điệp cũng đã kiểm tra xem thông điệp có phải DISCONNECT không. Việc ProcessQueueThread xử lý chỉ nhằm đảm bảo ServerProgram sẽ xóa ClientThread ra khỏi bộ nhớ.
* Khi ControlThread muốn ngắt kết nối đến toàn bộ client thì:
  + ConnectionHandlerThread sẽ được đóng trước, hầu như ngay lập tức do socket server sẽ bị đóng.
  + ControlThread sẽ lệnh DISCONNECT đến tất cả các clients, các ClientThread có thể được đóng.
  + ProcessQueueThread sẽ được đóng khi không còn request để xử lý.
* Khi tất cả các luồng con đã đóng, có thể đóng ControlThread.



1. Tập các lệnh có thể được sử dụng:

|  |  |  |
| --- | --- | --- |
| LỆNH TỪ CLIENT | CÂU TRẢ LỜI CÓ THỂ TỪ SERVER | Ý NGHĨA |
| LOGIN <username> <password> | SUCCEEDED | Đăng nhập thành công |
| FAILED USERNAME NOT FOUND | Đăng nhập thất bại, không tìm thấy tên người dùng |
| FAILED WRONG PASSWORD | Đăng nhập thất bại, mật khẩu không đúng |
| REGISTER <username> <password> | SUCCEEDED | Đăng ký thành công |
| FAILED USERNAME ALREADY EXISTS | Đăng ký thất bại, tên người dùng đã tồn tại |
| WEATHER ALL <date1> | SUCCEEDED <data> | Thành công Trả về dữ liệu là một chuỗi kí tự định dạng cấu trúc json |
| FAILED WRONG DATE FORMAT | Thất bại, <date> truyền không đúng định dạng |
| WEATHER RECENT <city\_id> <count> | SUCCEEDED <city\_name> <data> | Thành công  Trả về dữ liệu là tên thành phố và chuỗi kí tự định dạng cấu trúc json |
| FAILED NO CITYID | Thât bại, ID thành phố không đúng |
| DISCONNECT |  | Thông báo cho server rằng client sẽ kết nối.  Không trông đợi câu trả lời và client sẽ ngắt kết nối trong thời gian sớm nhất |
| <câu lệnh sai> | INVALID | Câu lệnh bị lỗi |
| LỆNH TỪ SERVER | CÂU TRẢ LỜI CÓ THỂ TỪ CLIENT | Ý NGHĨA |
| DISCONNECT | CONFIRM DISCONNECTION | Server cần ngắt kết nối và đang chờ client xác nhận |

Chú thích:

1. <date> có định dạng là ‘YYYY/MM/DD’
2. Cấu trúc gói tin thông điệp:

|  |  |  |  |
| --- | --- | --- | --- |
| Phần | HEADER | REQUEST\_ID | MESSAGE |
| Độ dài | 8 bytes | 2 bytes | X bytes |

* HEADER chứa độ dài của toàn bộ thông điệp (bao gồm cả HEADER) theo đơn vị byte.
* REQUEST\_ID là số thứ tự của thông điệp. Một yêu cầu của client và câu trả lời từ server cho nó phải có cùng REQUEST\_ID
* MESSAGE là bản thân thông điệp cần gửi

1. Tổ chức cơ sở dữ liệu:

*CƠ SỞ DỮ LIỆU NGƯỜI DÙNG:*

|  |
| --- |
| [          {  "admin\_username": "admin",           "admin\_password": "admin"          },  {  "username": "test1",           "password": "123"          },  {  "username": "test2",           "password": "456"          }  ] |

Cơ sở dữ liệu người dùng được lưu dưới dạng file json được trình bày dưới dạng một danh sách các tài khoản:

* Mỗi tài khoản có một tên đăng nhập(duy nhất) - username
* Ứng với mỗi tên đăng nhập có một mật khẩu - password
* Đặc biệt có một tài khoản của admin có tên đăng nhập/mật khẩu mặc định là admin/admin

*CƠ SỞ DỮ LIỆU THỜI TIẾT:*

|  |
| --- |
| {          "city\_id": 35687,          "city\_name": "Da Nang",          "date\_weather": [              {                  "date": "2021/05/19",                  "info": {                      "weather": "Sunny",                      "temperature": 30,                      "humidity": 0.6,                      "wind\_speed": 27.54                  }              },              {                  "date": "2021/05/20",                  "info": {                      "weather": "Raining",                      "temperature": 26,                      "humidity": 0.9,                      "wind\_speed": 37.54                  }              }          ] |

Cơ sở dữ liệu thời tiết được lưu dưới dạng file json được trình bày dưới dạng một danh sách các thành phố:

* Mỗi thành phố có một id duy nhất, được cấp phát tự động mỗi khi một thành phố mới được thêm vào.
* Mỗi thành phố có đi kèm một danh sách (date\_weather) ghi lại các dữ liệu thời tiết vào một ngày nhất định. Các ngày đều xác định và duy nhất theo định dạng ‘YYYY/MM/DD’
* Thông tin thời tiết bao gồm: Thời tiết dự báo (weather), nhiệt độ (temperature), độ ẩm (humidity), tốc độ gió (wind\_speed).

## Môi trường lập trình, framework:

Được viết trên Python 3.7.9

Chương trình sửa dụng các thư viện sau:

*Có sẵn:*

* threading: hỗ trợ xử lý đa luồng.
* socket: hỗ trợ API lập trình socket
* json: hỗ trợ đọc ghi file json.
* Các thư viện bổ trợ khác: datetime, logging, queue, os, pathlib, enum

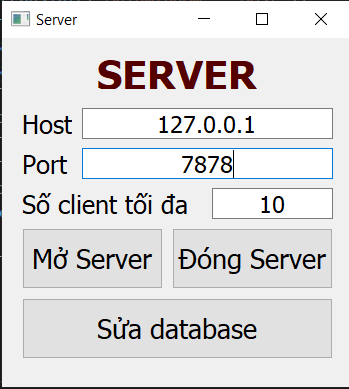
*Cài bổ sung:*

* pyqt5: hỗ trợ xây dựng GUI

## Hướng dẫn sử dụng: console-based module

+ Về phía Server:

Chạy file server\_gui.pyw, cửa sổ sau hiện ra

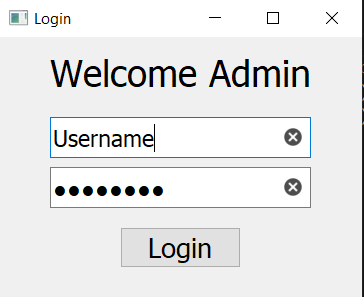


Nhập Host và số client tối đa vào các ô như hình trên

Nhấn nút Mở Server để thực hiện mở server (Port mặc định mở là 7878)

Nhấn nút Đóng Server để thực hiện đóng server

Nhấn nút Sửa database để thực hiện thêm, sửa, xóa thông tin thời tiết trong database đã có (lưu trong file json như đã đề cập ở trên), cửa sổ sau hiện ra

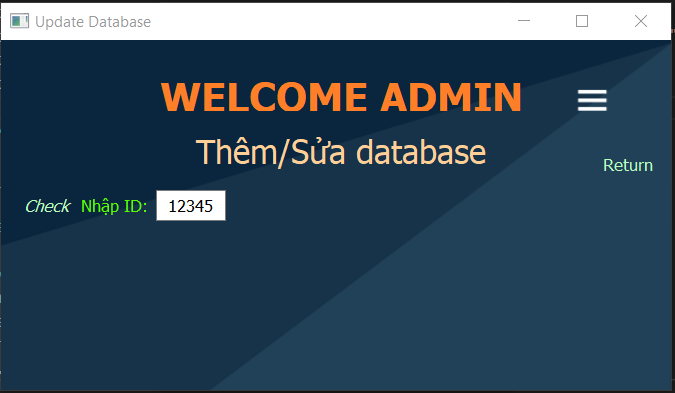


Cần đăng nhập vào tài khoản admin để truy cập vào database (tài khoản và mật khẩu mặc định là admin/admin), cửa sổ sau sẽ hiện ra



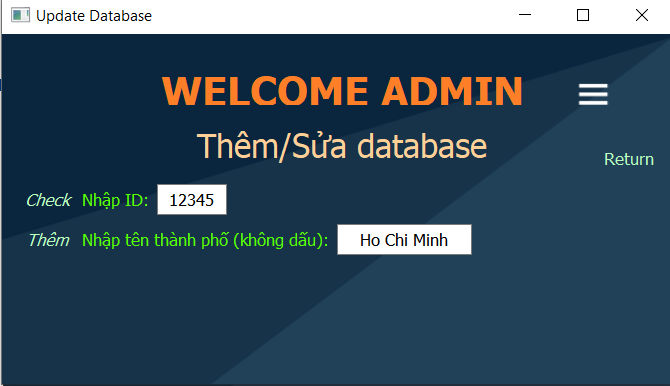
Có 3 chức năng như trong hình:

Chọn Thêm/Sửa Database: thêm thành phố, thông tin cho thành phố, cửa sổ sau sẽ hiện ra (Nút Return ở đây ấn vào sẽ quay lại màn hình ở trên)



Nhập ID thành phố vào ô trên sau đó ấn nút *Check:*

Nếu ID đã nhập chưa tồn tại thì **chức năng thêm thành phố** được chọn và cửa sổ sẽ hiển thị thêm ô để nhập tên thành phố



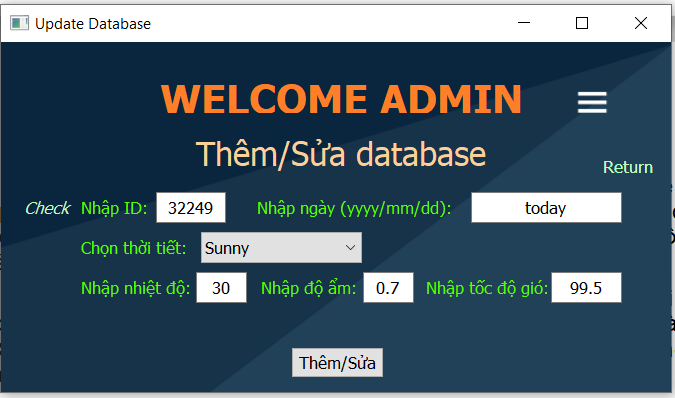
Nhập tên thành phố theo định dạng như trên và ấn nút *Thêm* để thêm thành phố, Khi đó thành phố được thêm kèm theo một ID, ID này sẽ được hiển thị ở ô trên và **chức năng thêm dữ liệu cho thành phố** được chọn (khi đó sẽ không có nút *Thêm* trên màn hình)

Trong trường hợp ID đã nhập tồn tại trong database rồi thì **chức năng thêm dữ liệu cho thành phố** được chọn và tên thành phố trong dữ liệu sẽ được hiển thị trong ô dưới.

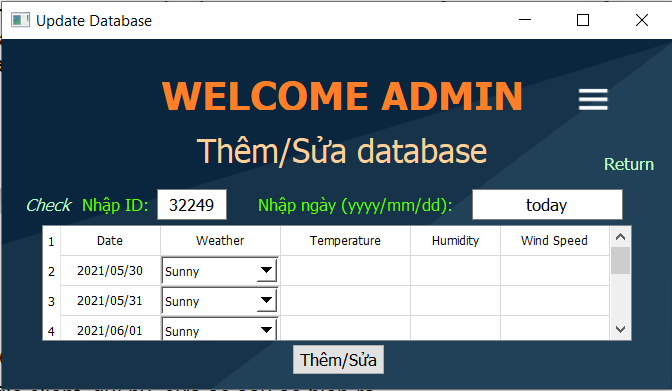
Sau đó cửa sổ sẽ hiện ra 2 nút để chọn chức năng thêm thông tin cho thành phố trong 1 ngày hoặc thêm thông tin cho 7 ngày liên tục. (Sau khi chọn 1 trong 2 chức năng được hiển thị thì nút Return trên màn hình sẽ đưa người dùng quay lại màn hình chọn 2 chức năng như dưới đây)



Với chức năng thêm thông tin cho thành phố trong 1 ngày, người dùng nhập các thông tin về thời tiết muốn thêm như giao diện dưới dây và bấm Thêm/Sửa (Nếu thành phố đó đã có thông tin thời tiết về ngày được chọn thì thao tác này sẽ sửa thông tin trong database)

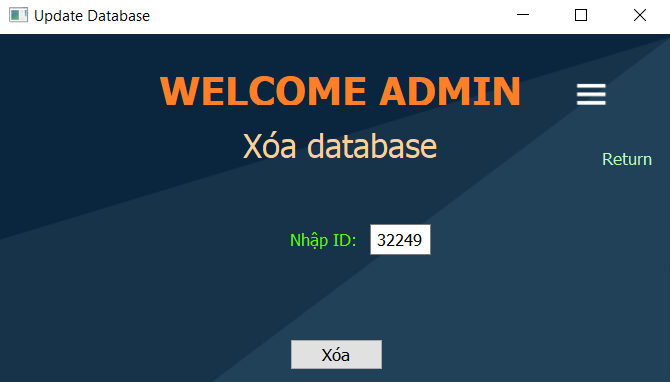


Với chức năng thêm thông tin cho thành phố trong 7 ngày liên tục, người dùng nhập vào ngày bắt đầu theo định dạng (có thể nhập “today” để tính từ hôm nay) vào ô dưới và nhấn nút *Thêm* sau đó nhập thông tin muốn thêm vào bảng hiện ra, hoàn thành nhập thông tin thì nhấn nút Thêm/Sửa để cập nhật database:

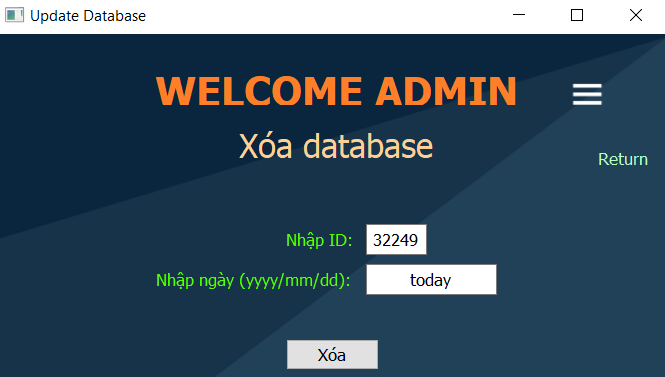


Lưu ý: Humidity (độ ẩm) phải có giá trị trong đoạn [0;1]

Chọn Xóa thông tin của thành phố để xóa đi mọi dữ liệu (bao gồm cả thành phố) trong database (cần nhập vào ID thành phố muốn xóa và chọn nút Xóa)

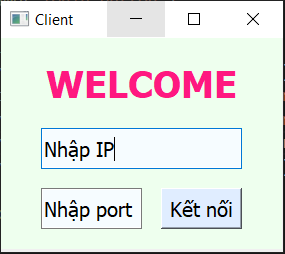


Chọn Xóa thông tin một ngày của thành phố để xóa đi dữ liệu thời tiết trong 1 ngày của 1 thành phố (cần nhập vào ID thành phố và ngày muốn xóa và chọn nút Xóa)

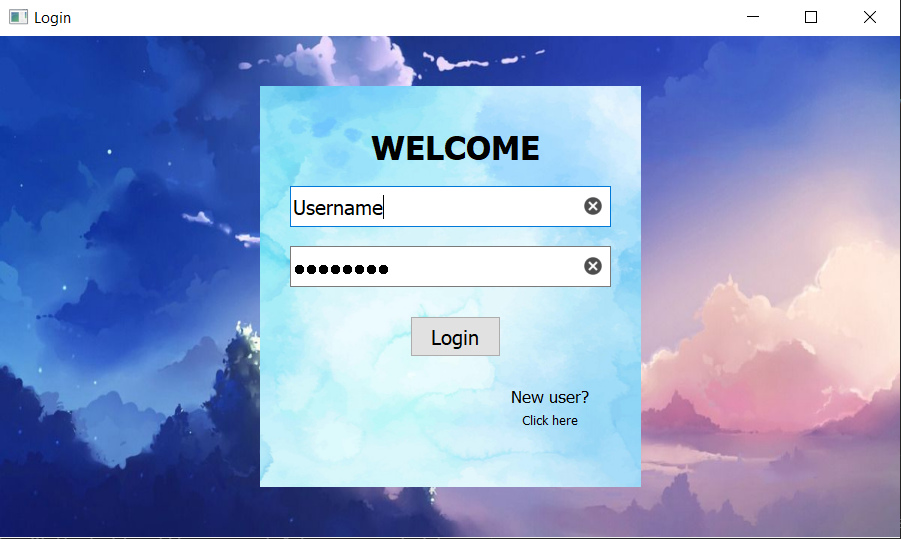


+ Về phía Client:

Chạy file client\_gui.pyw, cửa sổ sau sẽ hiện ra



Nhập vào IP và port của server muốn truy cập vào và nhấn nút Kết nối để giao tiếp với server. Nếu kết nối thành công, cửa sổ sau sẽ hiện ra



Nhập tên đăng nhập và mật khẩu để xem thông tin thời tiết. Nếu chưa có tài khoản thì có thể đăng ký(Làm theo hình). Đăng nhập thành công thì cửa sổ sau hiện ra



Chọn chức năng muốn thực hiện và tiếp tục làm theo hướng dẫn trong giao diện

Nút *Log out* trên góc phải màn hình để đăng xuất khỏi tài khoản và quay lại màn hình đăng nhập

## Bảng phân công:

|  |  |
| --- | --- |
| Công việc | Người thực hiện |
| Thiết lập mô hình giao tiếp | Trí |
| Lập trình giao tiếp client/server | Đức Anh |
| Lập trình hỗ trợ đa luồng phía server | Trí |
| Lập trình hỗ trợ đọc, giải quyết truy vấn thời tiết | Trí |
| Lập trình hỗ trợ giải quyết truy vấn người dùng (đăng nhập/đăng ký) | Đức Anh |
| Lập trình GUI | Đức Anh |

## Tài liệu tham khảo:

* 1. *Python official documentation: <https://docs.python.org/3/>*