# BÁO CÁO ĐỒ ÁN SOCKET WEATHER FORECAST

*Lưu ý: File .exe được cung cấp bởi link download ở mục 5*

Mục lục

[1. Thông tin nhóm: 1](#_Toc73302645)

[2. Đánh giá mức độ hoàn thành: 1](#_Toc73302646)

[3. Kịch bản giao tiếp của chương trình: 3](#_Toc73302647)

[4. Môi trường lập trình, framework: 8](#_Toc73302648)

[5. Hướng dẫn sử dụng: 8](#_Toc73302649)

[6. Bảng phân công: 17](#_Toc73302650)

[7. Tài liệu tham khảo: 17](#_Toc73302651)

## 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*** | Hoàn thành | Hoàn thành |
| **QUẢN LÝ KẾT NỐI** | Khi client hoặc server mất kết nối đột ngột, không làm chương trình treo hay xảy ra lỗi | Quản lý kết nối đa tiểu trình |
| ***Hoàn thành*** | Hoàn thành | Hoàn thành (Sử dụng số tiểu trình chưa tối ưu) |
| **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*** | Hoàn thành |  |
| **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*** | Hoàn thành |  |
| **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*** | Hoàn thành | Hoàn thành |
| **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*** | Hoàn thành | Hoàn thành |
| **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*** | Không thực hiện (dùng json) | Hoàn thành (dùng json) |
| **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*** | Hoàn thành | Hoàn thành |
| **GIAO DIỆN** | Console Application | Có thiết kế giao diện đồ hoạ cho chương trình (GUI) cho Client và Server |
| ***Hoàn thành*** | Không thực hiện | Hoàn thành (còn lỗi về giao diện không thống nhất) |
| ***Tổng*** | 95% | |

## Kịch bản giao tiếp của chương trình:

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

*PHÍA CLIENT*

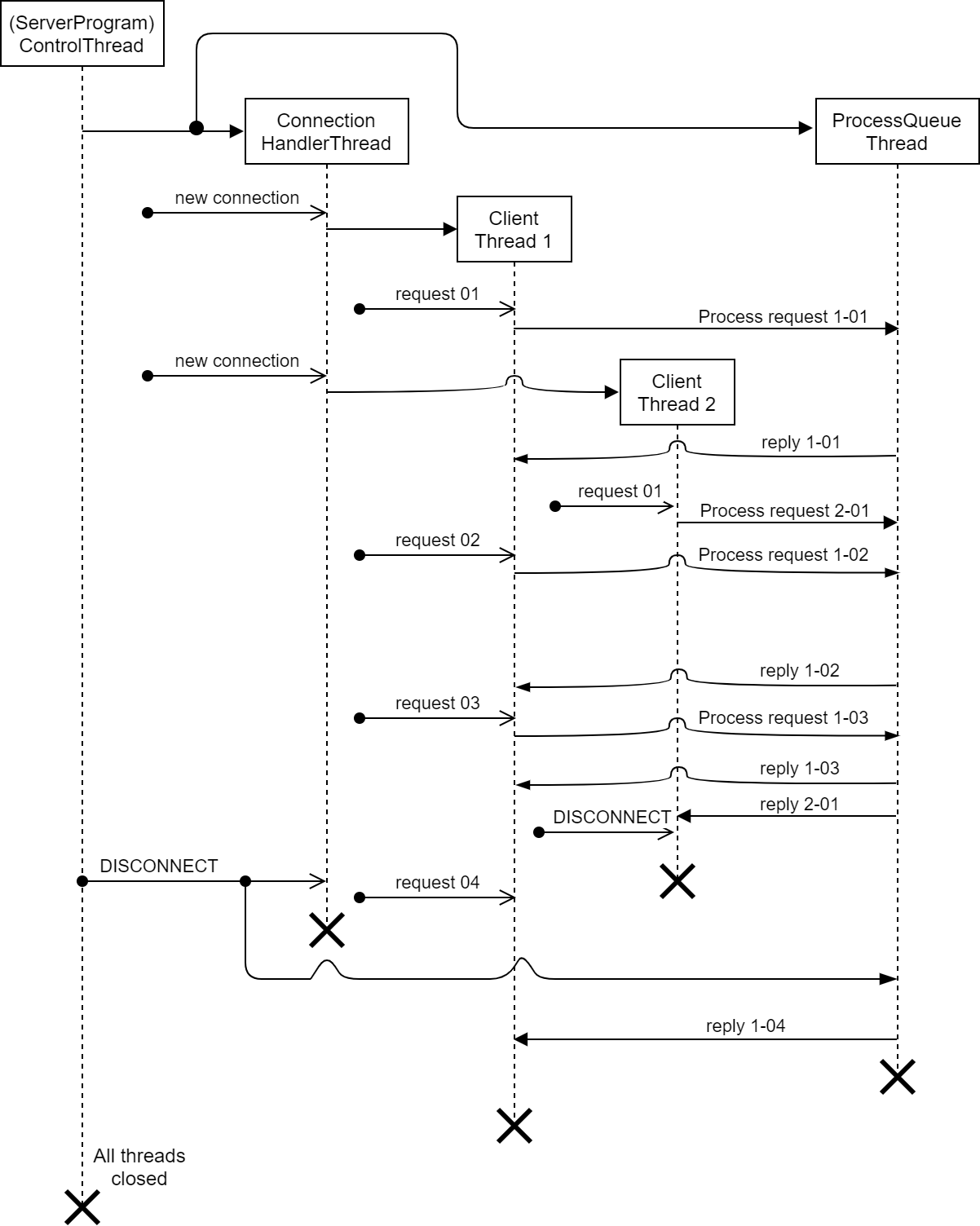
* Sau khi server mở kết nối, các client có thể kết nối đến host và port của server.
* Sau khi kết nối TCP được hình thành, client có thể:
  + Gửi yêu cầu DISCONNECT để đóng kết nối.
  + Gửi yêu cầu LOGIN để đăng nhập.
  + Gửi yêu cầu REGISTER để đăng ký.
* Nếu client đã đăng nhập thành công, client có 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 và kết thúc ngay nếu phải. 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 tự gửi lệnh DISCONNECT đến tất cả các clients, các ClientThread được đóng nhanh nhất có thể.
  + ProcessQueueThread sẽ được đóng khi không còn request để xử lý.
* Khi tất cả các luồng con đã đóng, có thể đóng ControlThread.

Sơ đồ ở trang bên mô tả tóm tắt một phiên làm việc của Server.



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 |
| FAILED WRONG VALUES | Thất bại, city\_id hoặc count sai định dạng |
| 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"          }  ] |

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.
* Một cơ sở dữ liệu người dùng chỉ có một tài khoản 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": "Rainy",                      "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/PySide2: hỗ trợ xây dựng GUI
* PyInstaller: hỗ trợ tạo exe

## Hướng dẫn cài đặt:

**Chạy file .exe**

Việc biên dịch ra một file .exe gọn nhẹ trên Python thuần khá phức tạp, do đó nhóm cung cấp một link để tải về (80mb): [dist.zip - Google Drive](https://drive.google.com/file/d/1LmPPkXIEbupMun6ci4bFq_WHL7fYee7O/edit)

**Chạy từ source**

*Yêu cầu: Python 3.7*.

1. Cài đặt các thư viện hỗ trợ bằng các lệnh sau đây:

|  |
| --- |
| > pip install PySide2 |

2. Chạy:

* file server\server\_gui.pyw để mở server
* file client\client\_gui.pyw để mở client.

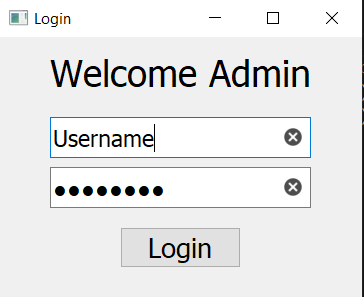
Bằng cách nhấn trực tiếp (file .pyw có thể chạy trực tiếp trên Windows) hoặc dùng cmd:

|  |
| --- |
| > python server\server\_gui.pyw  > python client\client\_gui.pyw |

## Hướng dẫn sử dụng:

***Server***

1. **Mở server để nhận kết nối**
   * Nhập IP host, port và số client tối đa.
   * Nhấn nút Mở Server để thực hiện mở server.
   * Nhấn nút Đóng Server để thực hiện đóng server.
2. **Cập nhật cơ sở dữ liệu thời tiết**

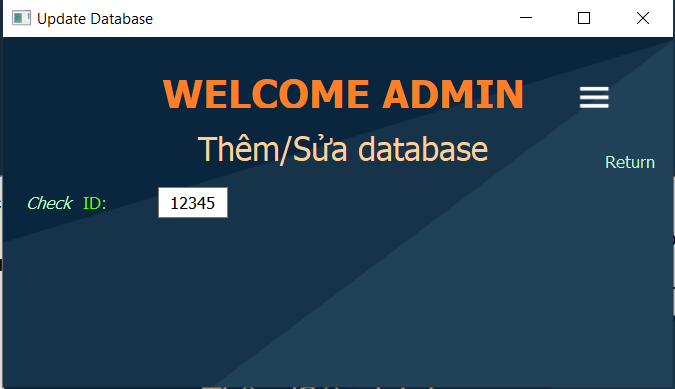
Cần đăng nhập vào tài khoản admin để truy cập vào database (tài khoản và mật khẩu mặc định là admin/admin).

Sau khi đăng nhập thành công thì cửa sổ sau sẽ hiện ra:

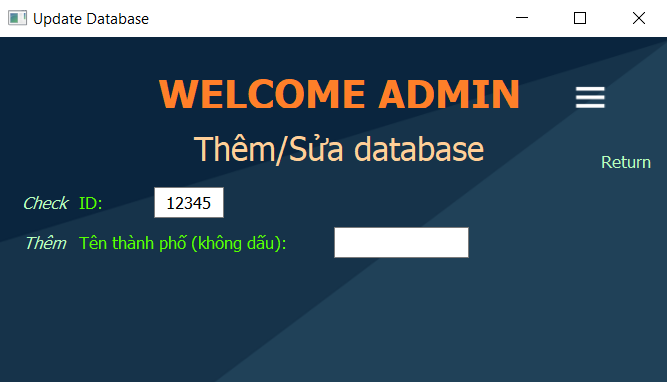


Có 3 chức năng như trong hình:

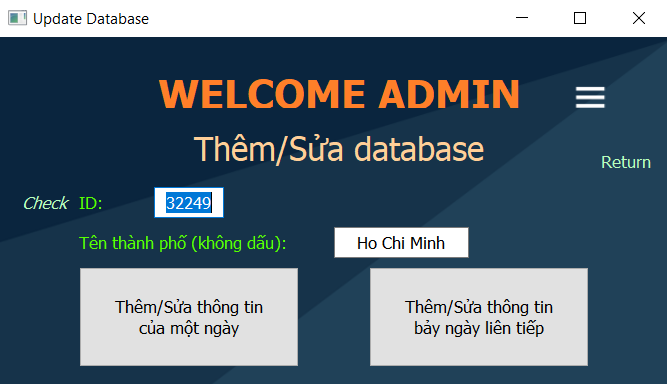
1. *Thêm/Sửa Database:*



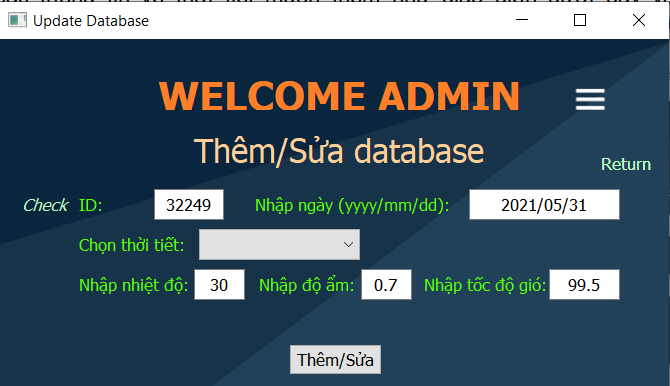
* Nhập ID thành phố vào ô trên sau đó ấn nút *Check.*
* Nếu ID đã nhập chưa tồn tại thì **chức năng thêm thành phố** được chọn và cửa sổ sẽ hiển thị thêm ô để nhập tên thành phố.



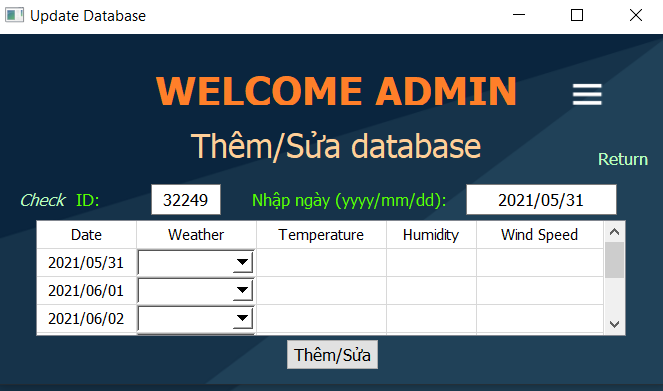
* Nhập tên thành phố và ấn nút *Thêm* để thêm thành phố, Khi đó thành phố được thêm kèm theo một ID **được cấp tự động**, ID này sẽ được hiển thị ở ô trên và **chức năng thêm dữ liệu cho thành phố** được chọn (khi đó sẽ không có nút *Thêm* trên màn hình)
* Trong trường hợp ID đã nhập tồn tại trong database rồi thì **chức năng thêm dữ liệu cho thành phố** luôn được chọn và tên thành phố trong dữ liệu sẽ được hiển thị trong ô dưới.



* Cửa sổ sẽ hiện ra 2 nút chức năng: *Thêm thông tin cho thành phố trong 1 ngày* hoặc *thêm thông tin cho 7 ngày liên tục*.
* 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. Dữ liệu đã tồn tại sẽ được ghi đè.

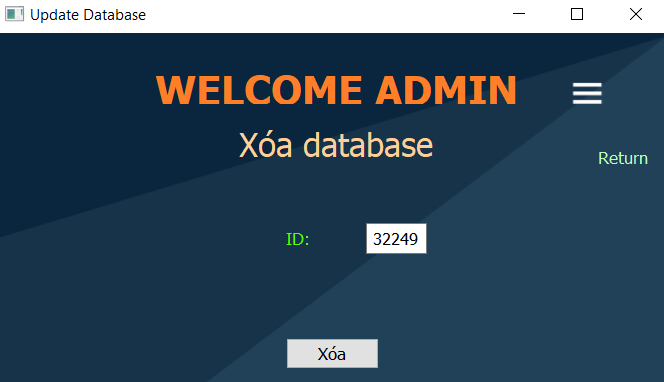


* Với chức năng *thêm thông tin cho thành phố trong 7 ngày liên tục*, người dùng nhập vào ngày bắt đầu theo định dạng vào ô dưới và nhấn nút *Thêm* sau đó nhập thông tin muốn thêm vào bảng hiện ra, hoàn thành nhập thông tin thì nhấn nút Thêm/Sửa để cập nhật database.



***Lưu ý:*** *Humidity (độ ẩm) phải có giá trị thực trong đoạn [0;1], Temperature và Wind Speed có giá trị thực.*

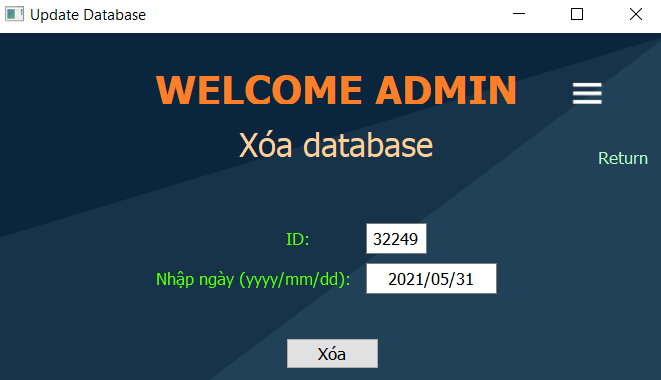
1. *Xóa thông tin của thành phố:*



* Nhập vào ID của thành phố và ấn nút xóa. Chỉ xóa thành công khi ID đó có trong cơ sở dữ liệu.

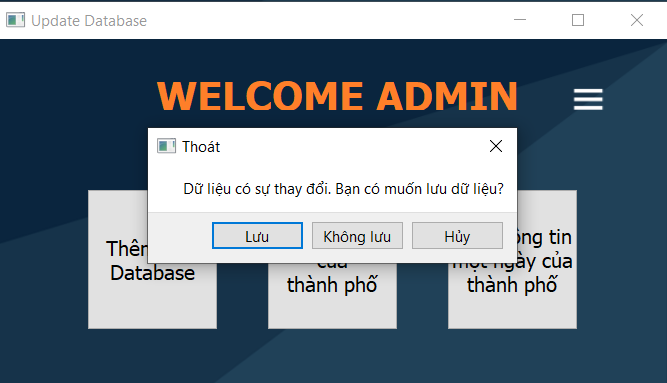
1. *Xóa thông tin một ngày của thành phố:*

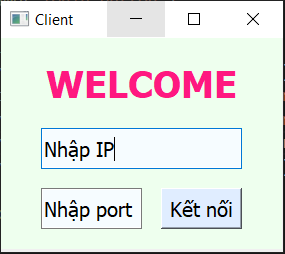
* Chọn Xóa thông tin một ngày của thành phố để xóa đi dữ liệu thời tiết trong 1 ngày của 1 thành phố (cần nhập vào ID thành phố và ngày muốn xóa và chọn nút Xóa)



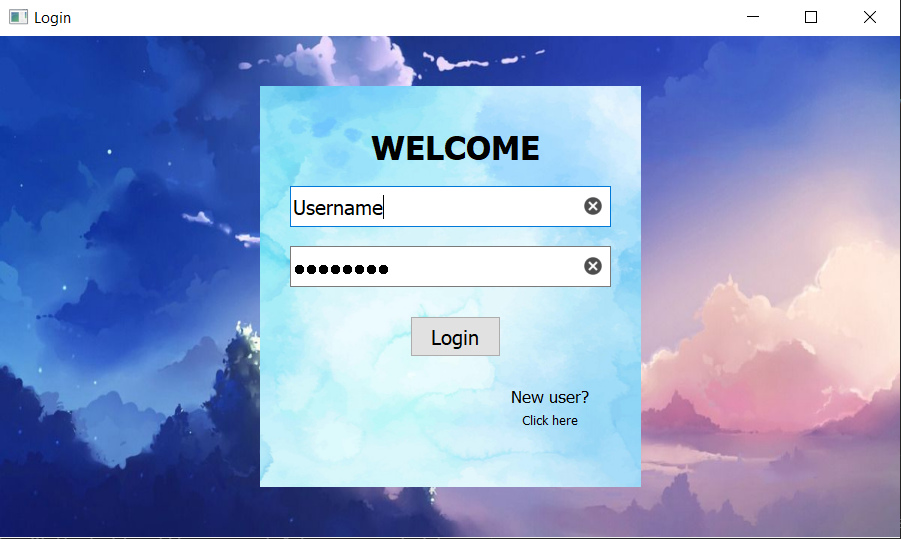
Nút Return được sử dụng để quy trở lại màn hình trước khi cần thiết.

Sau khi đã hoàn thành các thao tác cập nhật dữ liệu mong muốn thì người dùng chọn dấu X thoát cửa sổ Update Database và chọn Lưu (nếu muốn cập nhật database theo các thao tác đã làm) hoặc Không lưu (nếu không muốn sửa đổi database và bỏ qua các thao tác đã thực hiện ở trên).



***Client***

Nhập vào IP và port của server muốn truy cập vào (port mặc định là 7878) và nhấn nút Kết nối để thiết lập giao tiếp với server.

Nhập tên đăng nhập và mật khẩu để đăng nhập.

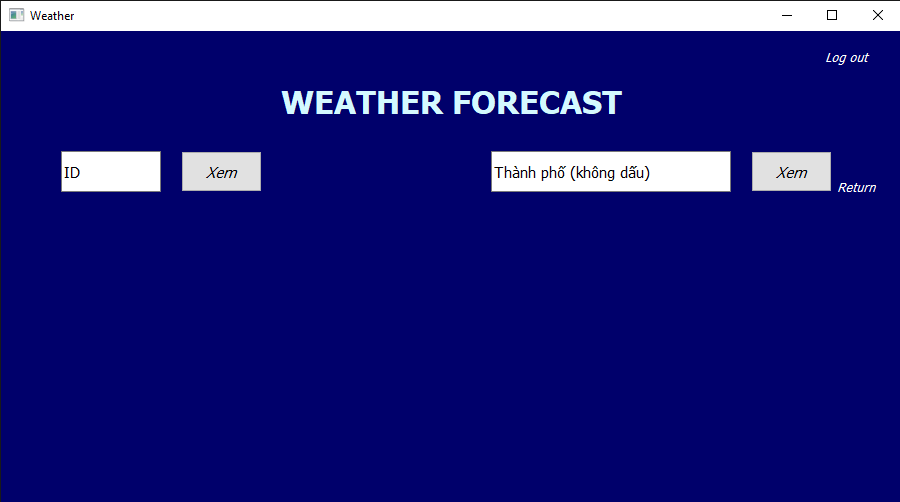
Nếu chưa có tài khoản thì có thể đăng ký bằng nút *New User: Click here*.



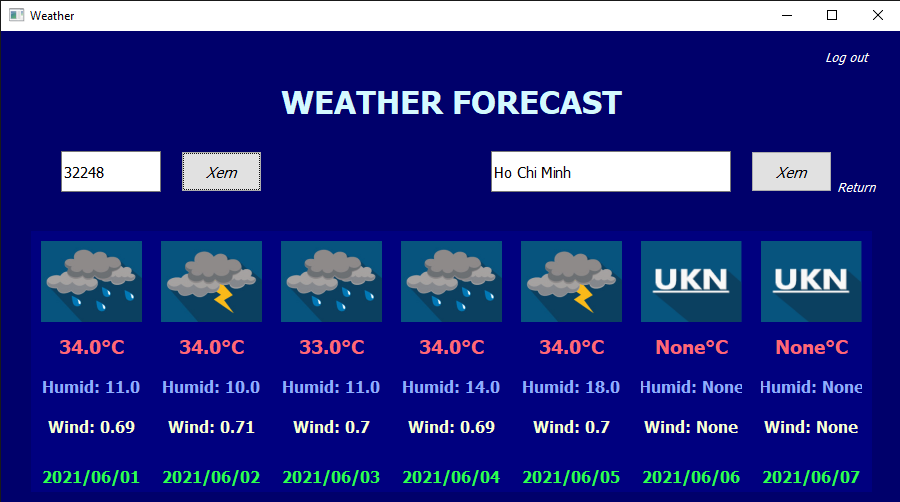
Có 2 chức năng như trong hình:

1. *Xem thông tin thời tiết một thành phố:*

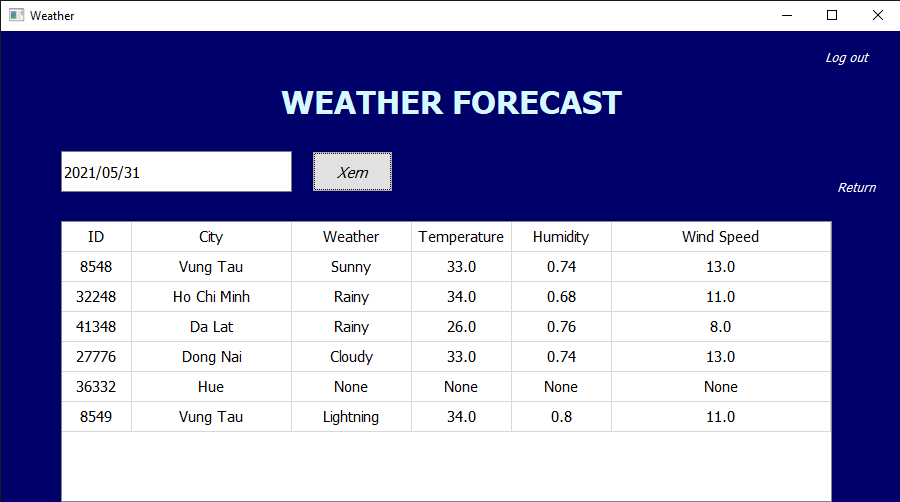
* Người dùng có thể lựa chọn xem bằng ID hoặc tên thành phố:
  + Đổi với ID, nhập ID thành phố vào ô “ID” bên trái. Nếu server tìm thấy thành phố, tên thành phố sẽ tự động điền vào ô bên phải.
  + Đổi với dùng tên, server không cung cấp tìm kiếm bằng tên, do đó chức năng này chỉ có thể sử dụng khi client có được một danh sách thành phố từ server (xem mục 2).



* Khi truy vấn thành công, kết quả sẽ là dự báo thời tiết cho bảy ngày tiếp theo.  
  Lưu ý: “UKN” nghĩa là server không có dữ liệu về ngày đó.



1. *Xem thông tin thời tiết các thành phố trong một ngày*



* Nhập ngày vào hộp thoại ngày theo định dạng ‘YYYY/MM/DD’ để lấy thông tin thời tiết của các thành phố.
* Thông tin về ID và tên thành phố sẽ được client tạm lưu để có thể sử dụng truy vấn thời tiết 7 ngày bằng tên thành phố thay cho ID.

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 cho server/client | 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/*](https://docs.python.org/3/)
  2. *Qt for Python* [Qt for Python — Qt for Python](https://doc.qt.io/qtforpython-5/)