

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG**



Báo cáo cuối kỳ môn lập trình mạng căn bản

A tournament football tracker



Giảng viên bộ môn:
Khán Hội

Sinh viên thực hiện:
Đỗ Phúc Kiên - 21522243
Ngô Vũ Minh Đạt – 21521935
Nguyễn Văn Đạt - 21521947
Lê Phi Hùng - 21522115

Lớp NT106.N22.MMCL - Khoá K16 - Nhóm 10

MỤC LỤC

1	TỔNG QUAN	3
1.1	Giới Thiệu	3
1.2	Lý do chọn đề tài	3
1.3	Cơ sở lý thuyết	3
1.4	Phân tích thiết kế hệ thống	4
1.4.1	Sơ đồ kiến trúc hệ thống	4
1.5	Network stack	5
1.6	Sơ đồ use case	6
1.7	Luồng ứng dụng	6
2	Hiện thực hệ thống	7
2.1	Chức năng tạo tài khoản, đăng nhập	7
2.2	Chức năng quên mật khẩu	9
2.3	Xem danh sách giải đấu, xem thông tin chi tiết của một cầu thủ, xem thông tin đội bóng	12
2.4	Triển khai trên Azure	14
2.5	Chức năng backup server	16
2.6	Chức năng xem highlight	17
2.6.1	Thành phần	17
2.6.2	Nguyên Lý hoạt động	17
2.6.3	Kết quả	17

1 TỔNG QUAN

1.1 Giới Thiệu

- “A tournament tracker” (trình theo dõi giải đấu) là một công cụ hoặc ứng dụng phần mềm được sử dụng để theo dõi tiến trình của một giải đấu. Nó thường được sử dụng trong thể thao, trò chơi và các sự kiện cạnh tranh khác, nơi nhiều người tham gia hoặc nhiều đội thi đấu với nhau trong một loạt các trận đấu hoặc trò chơi, cụ thể đề tài sẽ xây dựng một ứng dụng dùng để theo dõi một giải đấu bóng đá.
- Một trình theo dõi giải đấu bóng đá thường sẽ quản lý một số thông tin cơ bản về cầu thủ, về câu lạc bộ, về huấn luyện viên và một số thông tin liên quan khác. Ban tổ chức, cầu thủ và người dùng có thể sử dụng để cập nhật diễn biến giải đấu và theo dõi các cầu thủ hoặc đội bóng mà họ yêu thích.

1.2 Lý do chọn đề tài

- Việc lựa chọn đề tài "Tournament Tracker Football" là vô cùng hợp lý vì nó liên quan đến một trong những môn thể thao phổ biến nhất trên thế giới - bóng đá. Bóng đá không chỉ là môn thể thao giải trí mà còn là một phần của cuộc sống của hàng triệu người trên toàn thế giới, từ các cổ động viên trung thành đến những người chơi và các nhà quản lý.
- Với việc tổ chức các giải đấu và giải đấu bóng đá ngày càng phát triển, việc theo dõi kết quả của các đội bóng và các trận đấu trở nên cực kỳ quan trọng. Như vậy, việc xây dựng một hệ thống theo dõi giải đấu bóng đá sẽ giúp cho việc quản lý giải đấu và cập nhật thông tin trở nên dễ dàng hơn.
- Hơn nữa, đề tài "Tournament Tracker Football" còn giúp cho người dùng có thể theo dõi kết quả các trận đấu, thông tin về các đội bóng, cầu thủ, lịch thi đấu và nhiều thông tin khác liên quan đến giải đấu bóng đá. Điều này sẽ giúp cho người dùng có thể cập nhật thông tin nhanh chóng và thuận tiện hơn, đồng thời tăng cường sự quan tâm và niềm đam mê của người hâm mộ bóng đá.
- Vì vậy, việc nghiên cứu và phát triển một hệ thống theo dõi giải đấu bóng đá như "Tournament Tracker Football" sẽ có ý nghĩa rất lớn, không chỉ trong môi trường thể thao mà còn ở nhiều lĩnh vực khác như giáo dục, kinh doanh và truyền thông.

1.3 Cơ sở lý thuyết

Đề án này đã sử dụng những giao thức và phương thức sau:

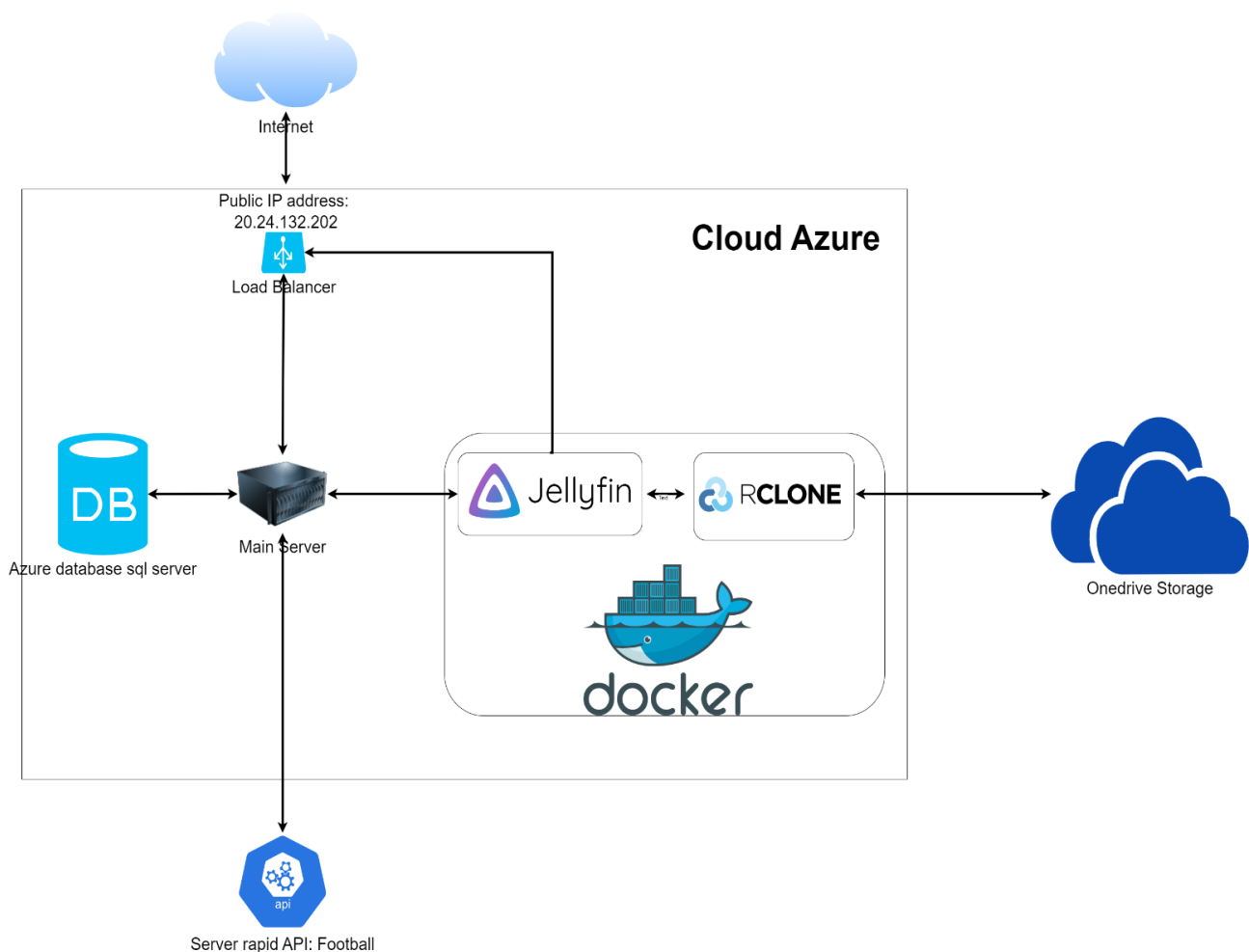
- TCP: Để đảm bảo độ tin cậy trong các ứng dụng mạng, ta dùng một giao thức có kết nối: TCP (Transport Control Protocol). Trên Internet chủ yếu dùng Telnet, HTTP, SMTP, POP3... Để lập trình theo giao thức TCP, .NET cung cấp hai lớp có tên là TCPClient và TCPListener. Trong đề án, TCP được sử dụng trong việc giao tiếp giữa Client và Server
- HTTP (Hypertext Transfer Protocol): một giao thức truyền tải siêu văn bản được sử dụng để truyền thông tin trên mạng. HTTP được sử dụng để tương tác giữa Client và Server, cho phép truyền tải và truy cập các tài liệu và dữ liệu trên World Wide Web. HTTP hoạt động dựa trên mô hình yêu cầu/phản hồi (request/response), trong đó máy khách gửi một yêu cầu HTTP tới máy chủ, và máy chủ phản hồi bằng một thông điệp phản hồi HTTP chứa thông tin được yêu cầu. HTTP là một giao thức cơ bản trong việc

truyền tải thông tin trên web và đóng vai trò quan trọng trong việc giao tiếp giữa các ứng dụng web và máy chủ. Trong đồ án, HTTP được sử dụng để gửi yêu cầu để client gửi yêu cầu đến server và jellyFin, Rclone cũng sử dụng API để lấy dữ liệu từ Onedrive

- API (Application Programming Interface): Một tập hợp các quy tắc, giao thức, công cụ. Dùng để xây dựng các ứng dụng phần mềm và cho phép các ứng dụng khác nhau giao tiếp và tương tác với nhau. Cho phép người dùng truy cập, sử dụng tài nguyên, dịch vụ từ một ứng dụng hoặc từ hệ thống khác. Dùng để truy cập dịch vụ web, cơ sở dữ liệu, thư viện mã nguồn mở, phần mềm hệ thống,... Được cung cấp dưới dạng các giao thức web, như REST, SOAP, JSON-RPC. Trong đồ án, API được sử dụng để server giao tiếp với Rapid API, Jellyfin.
- IP (Internet Protocol): Một giao thức trên Internet. IP là một địa chỉ đơn nhất. Ngày nay, IP được sử dụng để nhận diện và liên lạc với nhau trên mạng máy tính bằng cách sử dụng giao thức Internet. IP có hai loại chính: IPv4 và IPv6. IPv4 là phiên bản truyền thống được sử dụng rộng rãi, còn IPv6 là phiên bản mới hơn và hỗ trợ cho việc kết nối một số lượng lớn hơn các thiết bị trên mạng. Trong đồ án, IP được sử dụng để giao tiếp giữa Server và Client.
- Mã hóa bảo mật: ComputeSha256Hash là một phương thức tính toán giá trị băm (hash value) của một chuỗi dữ liệu sử dụng thuật toán băm SHA-256. SHA-256 (Secure Hash Algorithm 256-bit) là một thuật toán băm mã hóa đối xứng được sử dụng rộng rãi trong bảo mật thông tin và được coi là một trong những thuật toán bảo mật tốt nhất hiện nay. Phương thức ComputeSha256Hash thường được sử dụng trong các ứng dụng bảo mật, chẳng hạn như xác thực người dùng hoặc mã hóa mật khẩu. Khi một chuỗi dữ liệu được đưa vào phương thức này, nó sẽ tính toán giá trị băm của chuỗi đó bằng cách sử dụng thuật toán SHA-256 và trả về một chuỗi hexa (hệ cơ số 16) biểu diễn giá trị băm đó.

2 Phân tích thiết kế hệ thống

2.1 Sơ đồ kiến trúc hệ thống



Hình 1: Các thành phần triển khai trên Cloud Azure

Để hiện thực đồ án này thì có vài sự trợ giúp của các thành phần quan trọng sau:

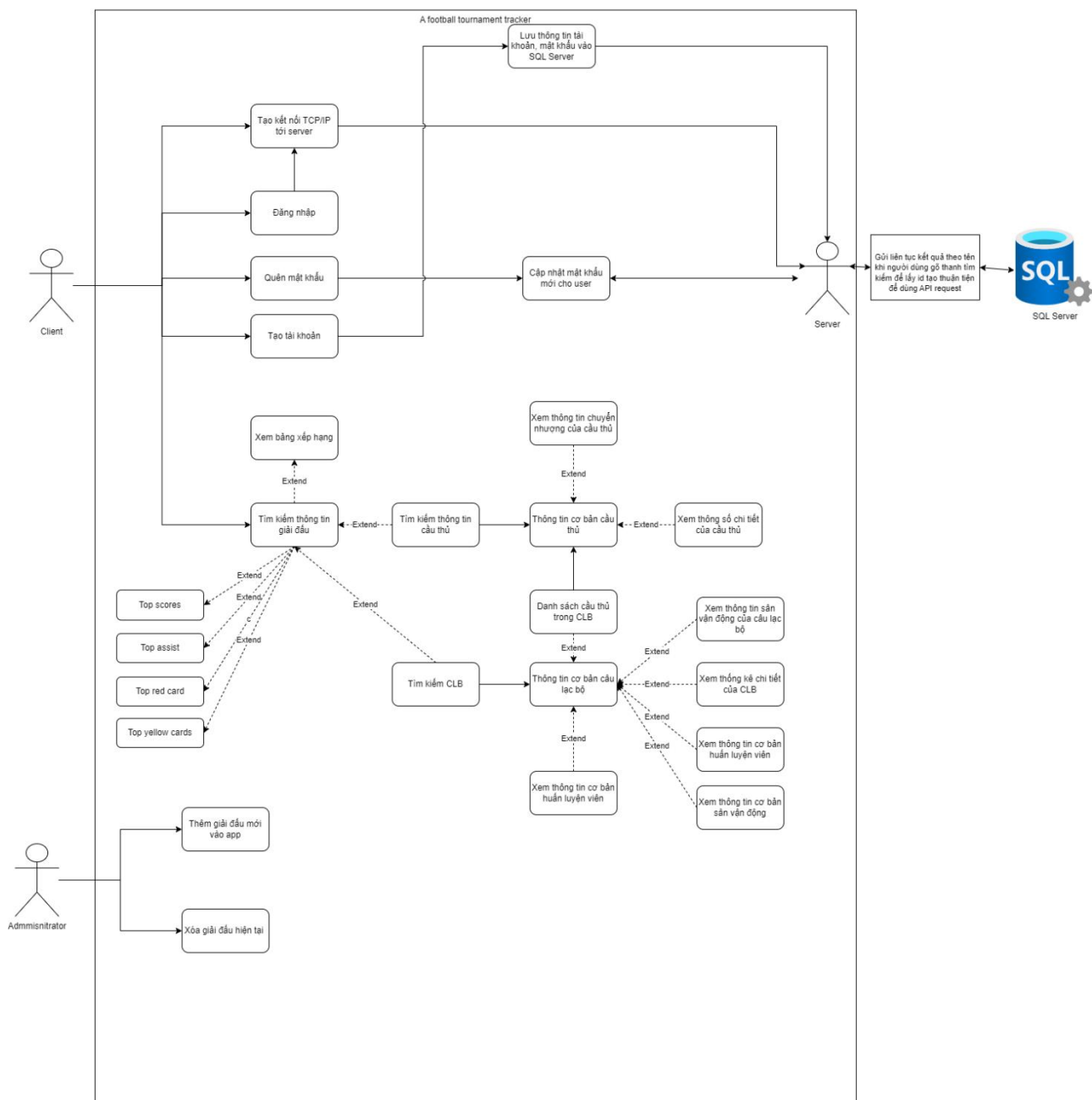
- Rapid API Football có trả phí: được sử dụng bên phía Main server để lấy thông tin từ server API.
- Ondrive Storage : là nơi lưu trữ video, chủ yếu được sử dụng cho chức năng streaming
- Rclone: là Một phần mềm mã nguồn mở, cho phép giao tiếp với Onedrive Storage thông qua API của họ.
- JellyFin là phần mềm Media Server mã nguồn mở, jellyfin sẽ đảm nhiệm chức cho chức năng streaming, JellyFin hỗ trợ API, và nó được sử dụng để giao tiếp với Main Server.
- Rclone và Jellyfin sẽ kết hợp lại giúp cho việc không cần video ở máy local mà có thể lấy từ Onedrive Storage và streaming cho client. Rclone và Jellyfin đã được config kĩ càng trong file compose-docker-file, sau đó chạy cảm file này trên nền tảng Docker của bất kì hệ điều hành nào.

2.2 Network stack

Đồ án này hoạt động trên các layer sau theo OSI model, bao gồm:

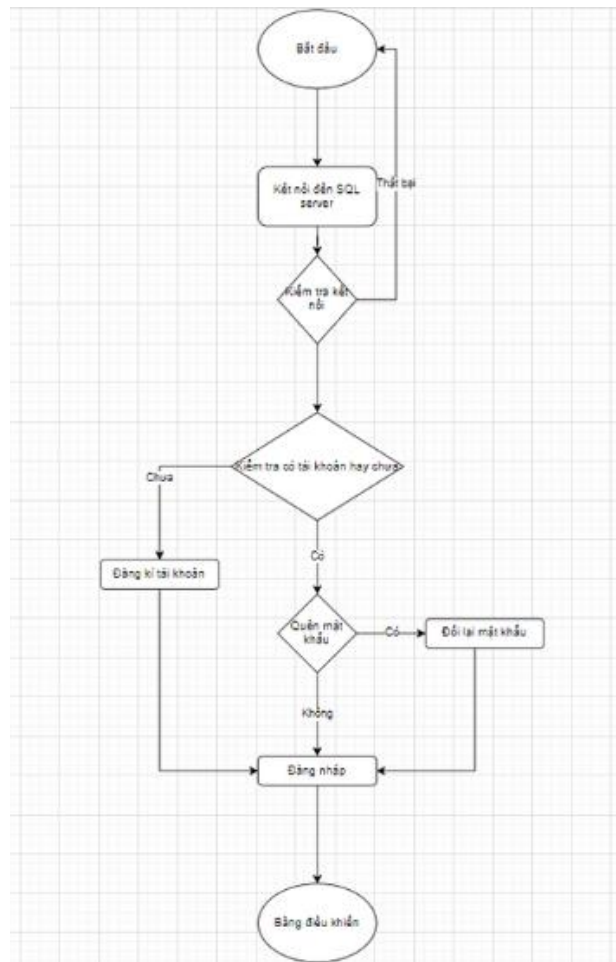
- Tầng Application: Sử dụng giao thức HTTP để gửi các yêu cầu HTTP đến API và nhận phản hồi từ API. Trong chức năng streaming của ứng dụng có sử dụng cả giao thức http để streaming video
- Tầng Presentation: Định dạng và mã hóa dữ liệu để đảm bảo tính toàn vẹn và bảo mật của thông tin truyền qua mạng.
- Tầng Transport: Sử dụng giao thức TCP để truyền dữ liệu đáng tin cậy giữa Server và Client.
- Tầng Network: Sử dụng giao thức IP để định tuyến gói tin và quản lý việc kết nối, giao tiếp giữa Server và Client.

2.3 Sơ đồ use case



Hình 2: Sơ đồ use case tổng quát của ứng dụng

2.4 Luồng ứng dụng



Hình 3: Sơ đồ use flow

3 Hiện thực hệ thống

3.1 Chức năng đăng ký , đăng nhập

Chức năng đăng ký tài khoản sẽ giúp người dùng có thể tạo một tài khoản mới, tài khoản đăng ký sẽ bao gồm các thuộc tính: “Tên tài khoản”, “Mật khẩu”, “Email”> Các thuộc tính này sẽ được lưu trong SQL User. Nguyên tắc hoạt động vô cùng đơn giản, đầu tiên ở bước đăng nhập có một nút “Đăng ký”. Khi nhấn vào nút này, người dùng sẽ được chuyển đến một trang đăng ký để có thể thực hiện đăng ký tài khoản. Sau khi đã có tài khoản thì người dùng sẽ có thể đăng nhập bằng tài khoản đã đăng ký. Nếu như đã có tài khoản, người dùng sẽ sử dụng tên đăng nhập và mật khẩu đã đăng ký để có thể đăng nhập. Ngoài ra còn có nút “Quên mật khẩu” để người dùng có thể tiến hành lấy mã xác minh tài khoản thông qua địa chỉ gmail.

The image shows a software window titled "Client" with a yellow background. At the top, there is a circular profile picture placeholder containing an illustration of a man with orange hair, sunglasses, and a red striped shirt. Below the profile picture are two input fields: the first is preceded by a black person icon, and the second is preceded by a black padlock icon. To the right of the second input field is a button labeled "Quên mật khẩu" (Forgot password). At the bottom of the window are two buttons: "Đăng ký" (Register) on the left and "Đăng nhập" (Login) on the right.

Hình 4: Form đăng nhập

Sau khi nhấn nút “Đăng ký”, người dùng sẽ được chuyển đến form tiếp theo

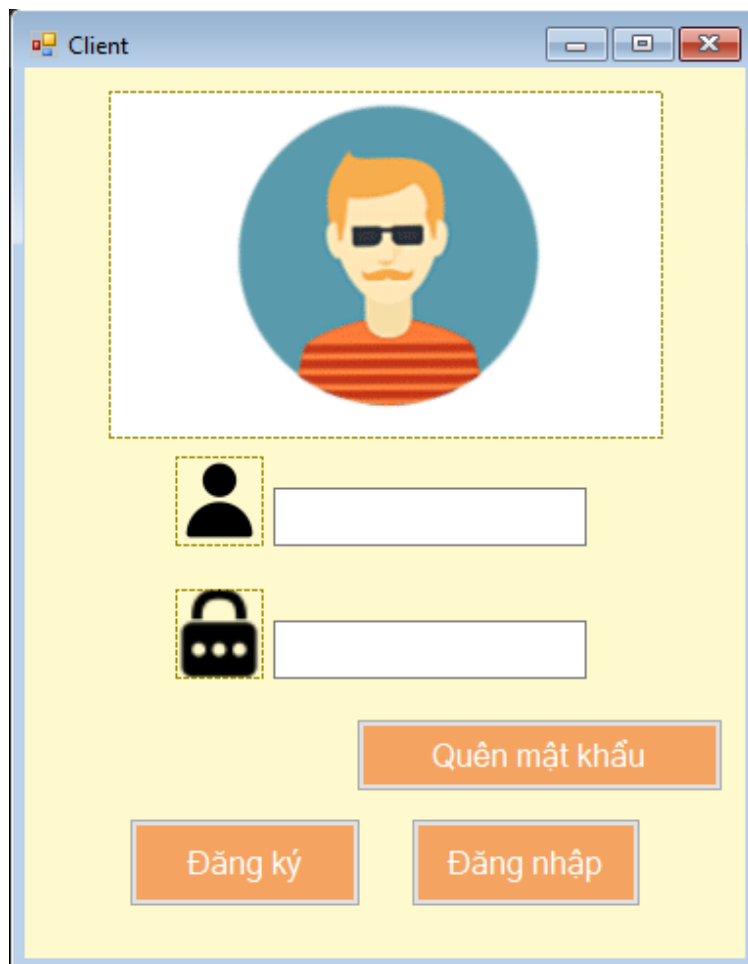
Hình 5: Form đăng ký

Ở form này hệ thống sẽ yêu cầu người dùng điền đầy đủ 3 thuộc tính “Tên tài khoản”, “Mật khẩu” và “Email”, sau đó người dùng ấn vào nút “Đăng ký”. Lúc này, hệ thống sẽ kiểm tra một lượt SQL User xem tài khoản này đã tồn tại hay chưa, nếu chưa thì tài khoản này sẽ được lưu vào bảng SQL User và hệ thống sẽ thông báo đăng ký tài khoản thành công rồi chuyển người dùng về lại form đăng nhập, còn nếu tài khoản này đã tồn tại rồi thì hệ thống sẽ báo tài khoản này đã tồn tại và yêu cầu người dùng nhập một tài khoản mới.

Đối với chức năng đăng nhập, người dùng sẽ nhập “Tên tài khoản” và “Mật khẩu” rồi nhấn nút đăng nhập. Lúc này, hệ thống sẽ duyệt qua SQL User một lượt và xem tài khoản và mật khẩu này đã tồn tại hay chưa, nếu tài khoản không tồn tại hoặc người dùng nhập sai tài khoản hoặc mật khẩu, hệ thống sẽ thông báo tài khoản không hợp lệ, từ đó người dùng có thể đưa ra lựa chọn, hoặc kiểm tra lại tài khoản mật khẩu một lần nữa rồi đăng nhập lại, hoặc nhấn nút “Đăng ký” để tạo ra một tài khoản mới, hoặc nhấn nút “Quên mật khẩu” để thay đổi mật khẩu. Còn nếu tên tài khoản và mật khẩu đã được nhập chính xác, phía máy chủ của người dùng sẽ gửi yêu cầu đăng nhập đến Server, và Server sẽ xử lý việc đăng nhập cho phía máy khách. Đồng thời, màn hình phía Server sẽ hiển thị nhận kết nối từ ip phía máy khách.

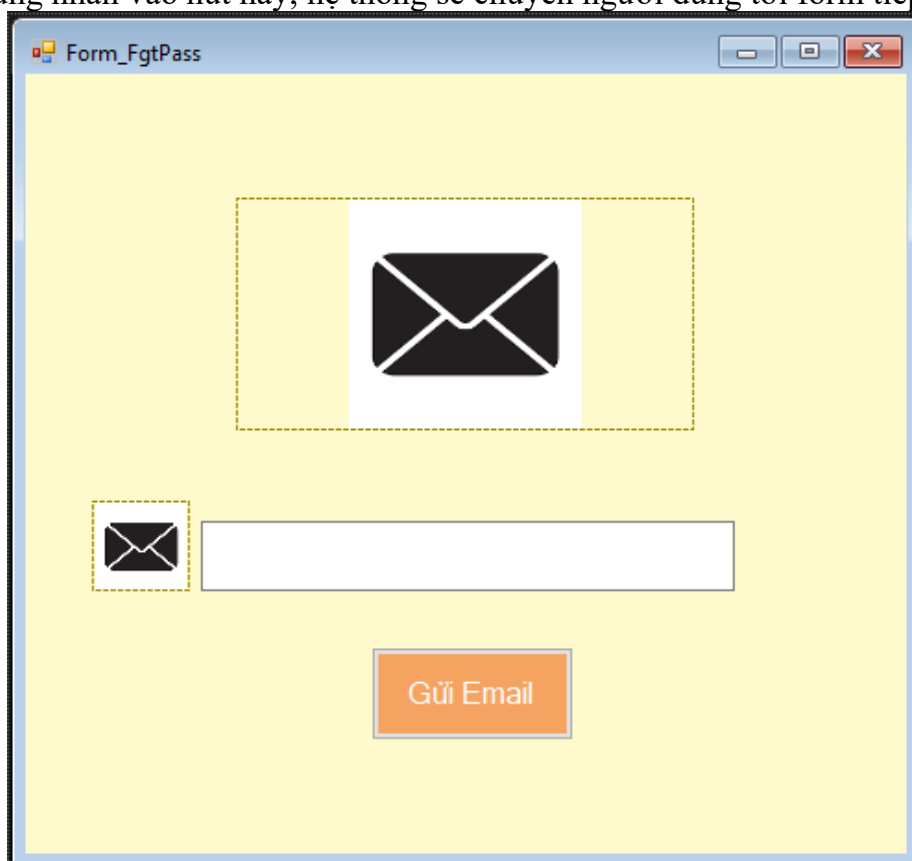
3.2 Chức năng quên mật khẩu

Chức năng quên mật khẩu này sẽ giúp người dùng có thể cập nhật mật khẩu khi vô tình quên mật khẩu cũ và mật khẩu mới này sẽ được cập nhật trong SQL User. Nguyên tắc hoạt động vô cùng đơn giản, đầu tiên ở bước đăng nhập có một nút “Quên mật khẩu”



A screenshot of a Windows application window titled "Client". The window has a yellow background and a blue title bar with standard Windows window controls (minimize, maximize, close). At the top center, there is a circular profile picture placeholder containing a cartoon illustration of a man with orange hair, sunglasses, and a red striped shirt. Below the profile picture, there are two input fields. The first input field is preceded by a person icon, and the second input field is preceded by a padlock icon. To the right of the second input field is an orange button labeled "Quên mật khẩu" (Forgot password). Below the input fields, there are two orange buttons: "Đăng ký" (Register) on the left and "Đăng nhập" (Login) on the right.

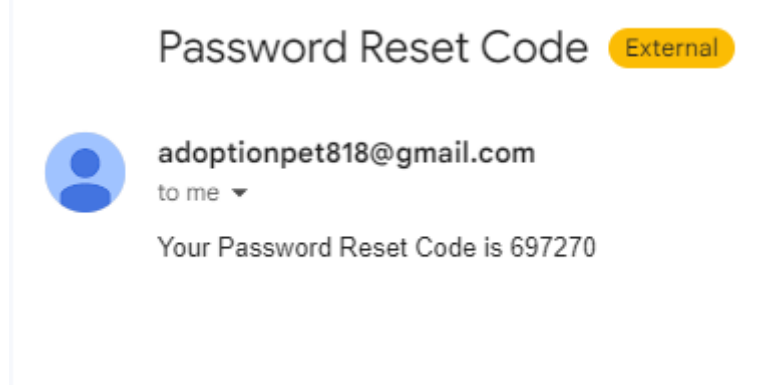
Khi người dùng nhấn vào nút này, hệ thống sẽ chuyển người dùng tới form tiếp theo



A screenshot of a Windows application window titled "Form_FgtPass". The window has a yellow background and a blue title bar with standard Windows window controls (minimize, maximize, close). At the top center, there is a large square placeholder containing a black envelope icon. Below this placeholder, there is a smaller square placeholder containing a black envelope icon, followed by a long white input field. At the bottom center of the window, there is an orange button labeled "Gửi Email" (Send Email).

Hình 6: Form nhập email để lấy lại mật khẩu

Ở bước này, người dùng sẽ nhập vào Textbox Email mà người dùng đã sử dụng để đăng ký tài khoản, sau đó nhấn nút “gửi Email”. Lúc này hệ thống sẽ kiểm tra xem Email đó có tồn tại trên SQL User hay không ,nếu không hệ thống sẽ báo không tìm thấy email, còn nếu email có tồn tại trên SQL User, hệ thống sẽ gửi một mã OTP về Email mà người dùng vừa gửi đi, mã OTP này dưới dạng 6 chữ số ngẫu nhiên. Sau đó người dùng sẽ bật Email mình lên để xem mã OTP đã được gửi về máy



Hình 7: Mã OTP được gửi vào mail sau khi đã xác minh thành công địa chỉ gmail
Sau khi nhấn nút “gửi Email”, hệ thống sẽ tiếp tục chuyển ta đến một form mới

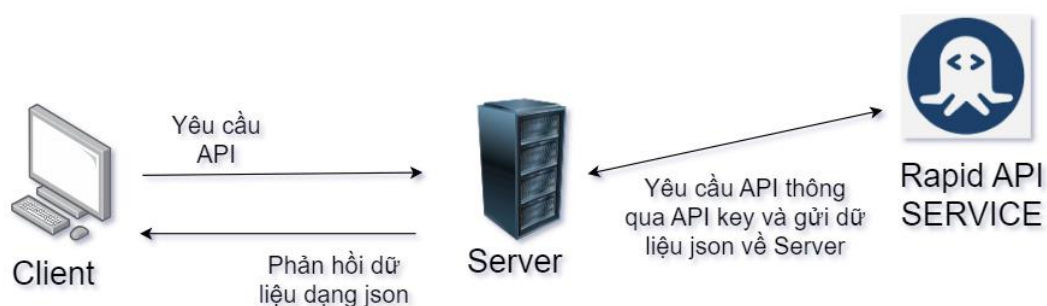
A screenshot of a web browser window titled "Form_Verify". The background is light yellow. In the center, the text "Nhập mã OTP" is displayed in large orange font. Below this text is an icon of an envelope with a keyhole, representing an email. To the right of the icon is a white rectangular text input field. Below the input field is an orange button with the text "Xác thực" in white.

Người dùng sẽ nhập mã OTP nhận được ở email vào TextBox và bấm nút “Xác thực”. Lúc này, hệ thống sẽ kiểm tra xem mã OTP người dùng vừa nhập có đúng với mã OTP mà hệ thống đã gửi tới email người dùng trước đó hay không, nếu không hệ thống sẽ báo mã OTP không phù hợp, còn nếu đúng hệ thống sẽ chuyển người dùng đến với form tiếp theo.

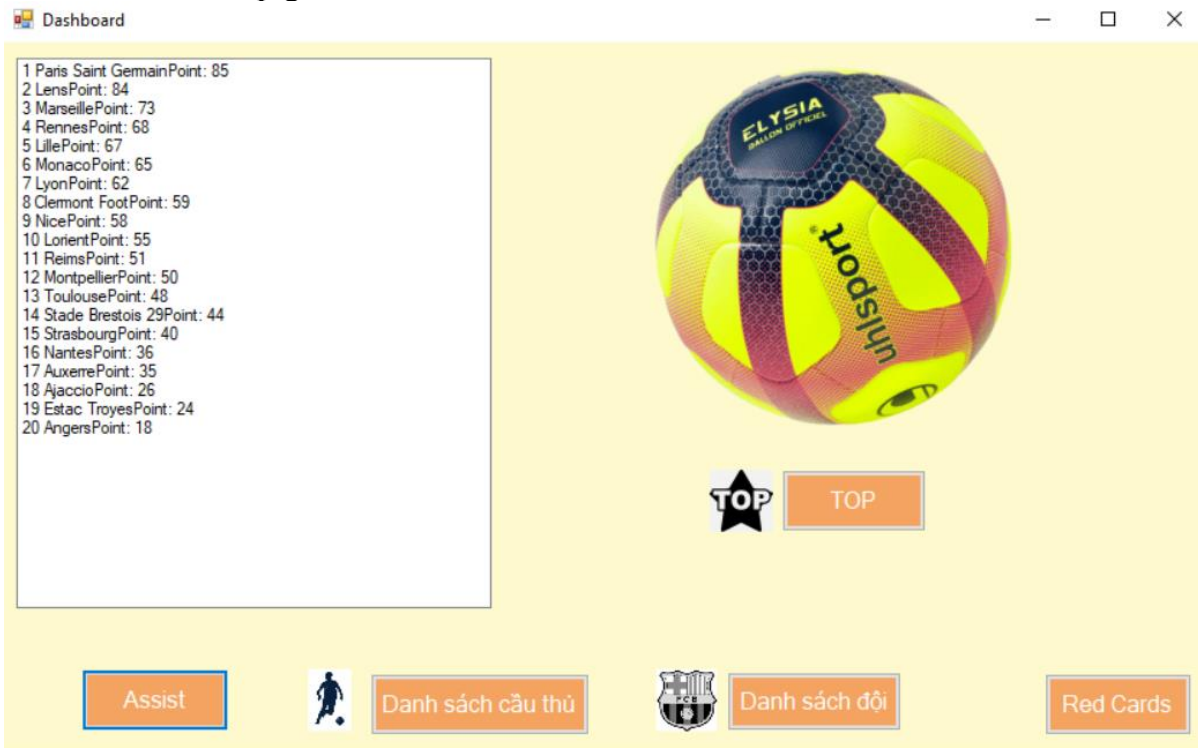
Người dùng sẽ nhập mật khẩu mới và nhập lại mật khẩu lần nữa để xác thực mật khẩu, sau đó người dùng sẽ bấm nút “Xác thực”. Lúc này hệ thống sẽ kiểm tra xem mật khẩu vừa nhập và mật khẩu xác thực có giống nhau không, nếu khác nhau hệ thống sẽ báo để người dùng có thể sửa lại, còn nếu giống nhau hệ thống sẽ thông báo mật khẩu được cập nhật thành công và mật khẩu mới sẽ được thay thế mật khẩu cũ trong cột “password” trong SQL User. Sau đó hệ thống đưa người dùng quay trở lại form đăng nhập.

3.3 Xem danh sách giải đấu, xem thông tin chi tiết của một cầu thủ, xem thông tin đội bóng

Chức năng xem danh sách giải đấu sẽ được vận hành theo mô hình Client-Server, giao tiếp thông qua giao thức TCP. Đầu tiên, Client sẽ yêu cầu một nội dung bất kỳ để hiển thị lên Dashboard, cho nên sẽ gửi một yêu cầu lên Server. Sau khi Server nhận được yêu cầu, Server sẽ gửi một yêu cầu API trên một dịch vụ miễn phí (rapidAPI) để lấy về nội dung dưới dạng json. Server nhận được dữ liệu, dữ liệu sẽ được gửi về phía Client để xử lý và hiển thị dữ liệu.



Giao diện tương tác của Client, bao gồm các nút chức năng tìm kiếm, hiển thị danh sách cầu thủ, danh sách đội, top ghi bàn, thẻ đỏ, hỗ trợ.




Khi user ấn vào các button, client sẽ yêu cầu gửi thông tin nội dung cần hiển thị. Ở bên phía server nhận được yêu cầu, cập nhật trạng thái:



Sau đó server gửi dữ liệu về cho Client và hiển thị trên giao diện:

Team_form



Logo

Thông tin cơ bản

Tên đội: Monaco

Kí hiệu: MON

Thành lập: 1919

Quốc gia: France

Đội tuyển quốc gia: False

Sân vận động: Stade Louis II

Sức chứa: 18523

Địa chỉ: 7, avenue des Castelans

Thành phố: Monaco

Danh sách đội

Montpellier

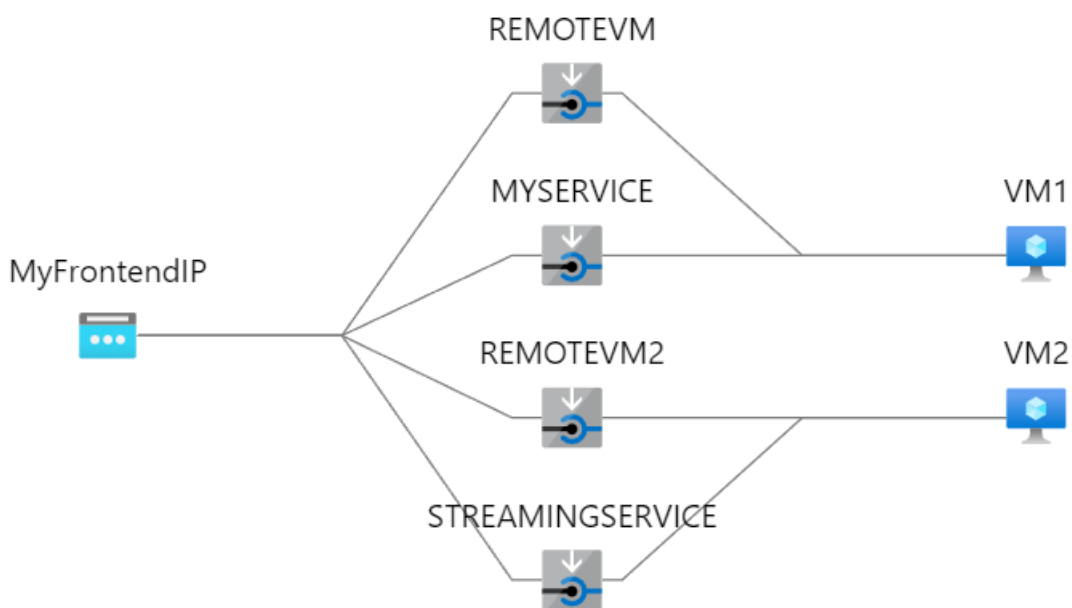
Monaco

Mo

Tim đội

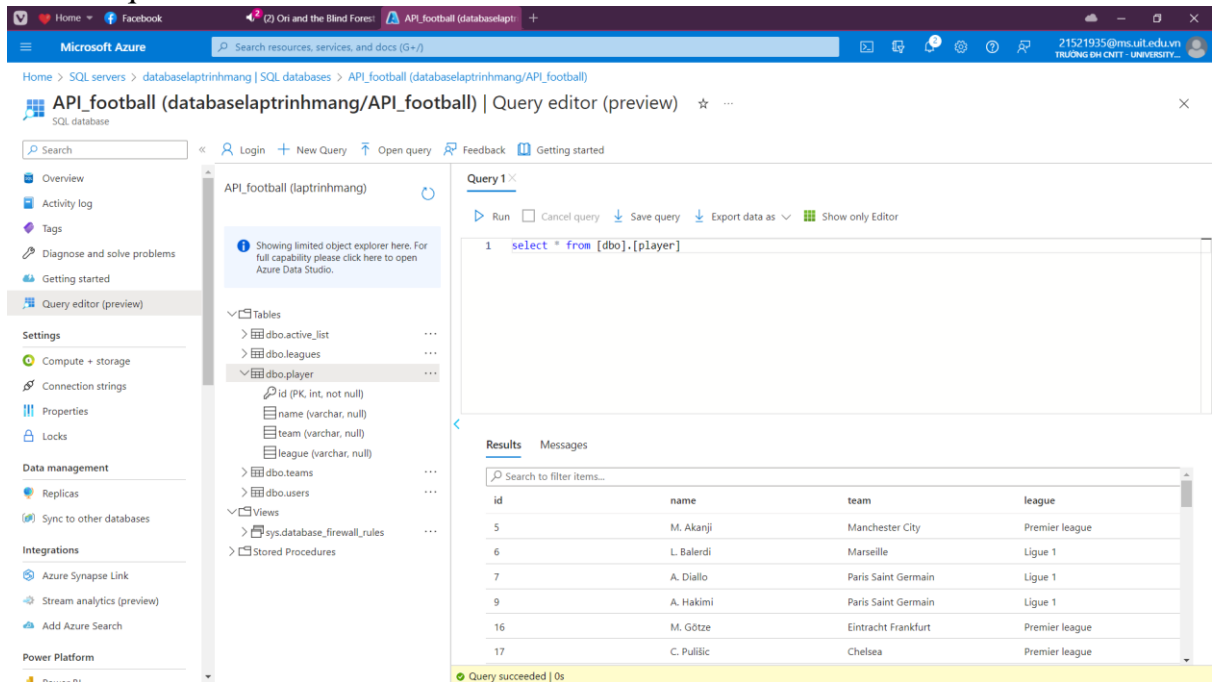
Thông tin chi tiết

3.4 Triển khai trên Azure



Hình 2. Mô hình loadbalancer với các nat rules

- 2 máy ảo window
- Azure Sql server



Hình 3. Sql Server

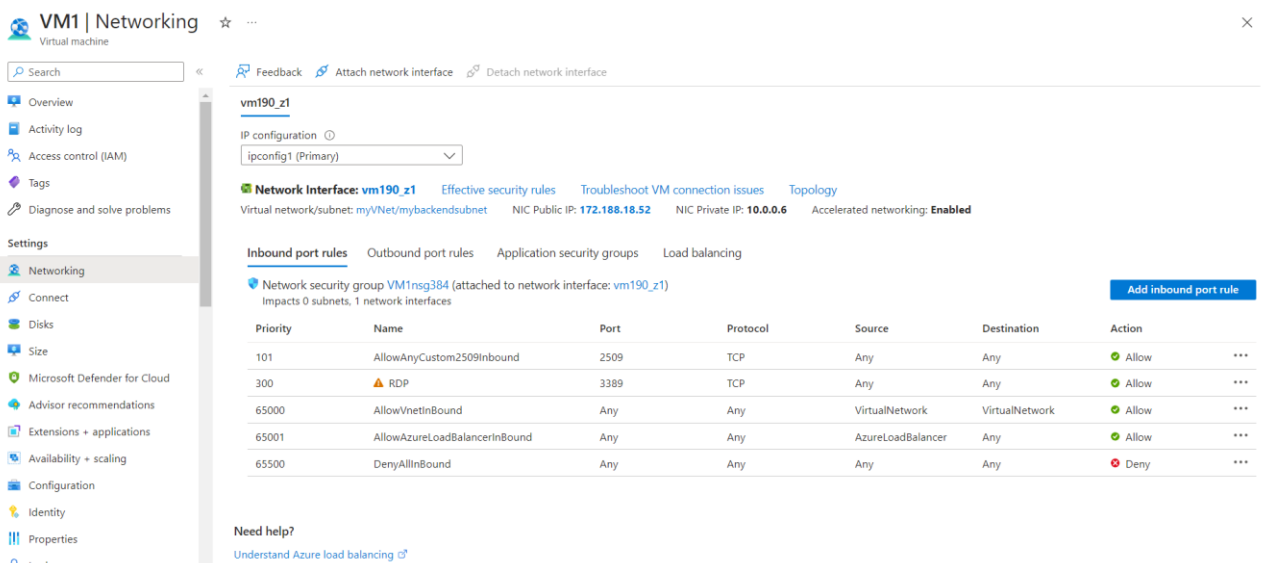
- Load balancer đã được cấu hình inbound nat rules port 2509 (main server) cho Vm1 và port 8096 cho VM2 (Jellyfin)

+ Add Refresh Give feedback

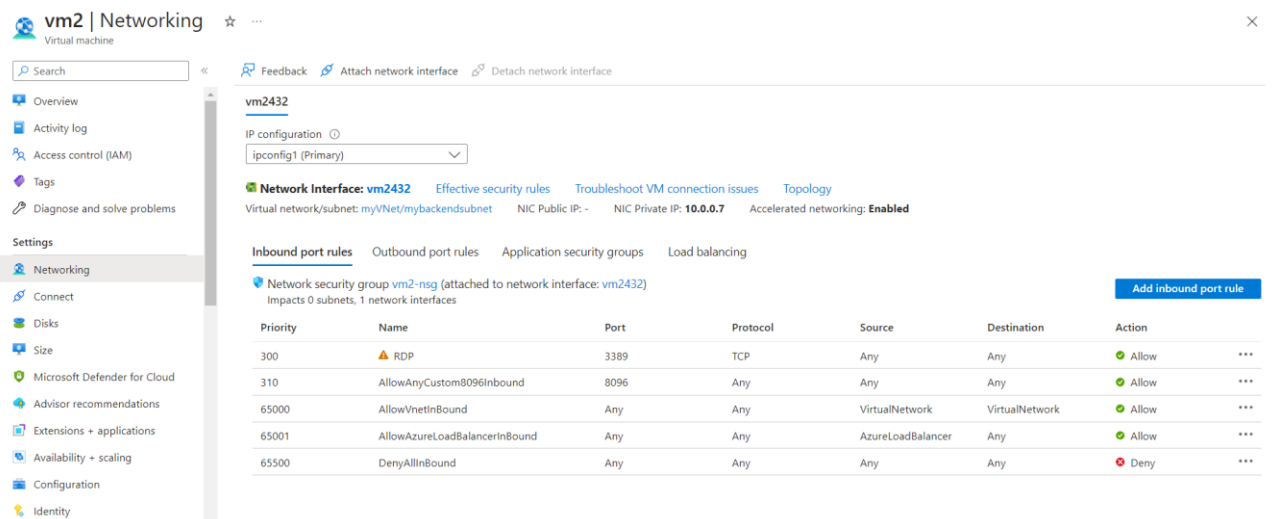
Filter by name...				
Name ↑↓	Frontend IP ↑↓	Frontend port/range ↑↓	Target ↑↓	Service ↑↓
remoteVM	20.24.132.202	3389	VM1	RDP (TCP/3389) ...
MyService	20.24.132.202	2509	VM1	Custom (TCP/2509) ...
remotem2	20.24.132.202	3390	vm2	RDP (TCP/3389) ...
streamingservice	20.24.132.202	8096	vm2	Custom (TCP/8096) ...

Hình 4. Cấu hình trên Inbound nat rules của Loadbalancer, 3389 để remote VM1, 3390 cho VM2

- Vm2 sử dụng docker để chạy 2 dịch vụ Jellyfin và rclone
- Cấu hình cho phép các port vừa đề cập vượt qua tường lửa của Azure



Hình 5. Inbound port rules của Vm1, cho phép port 2509 cho main server



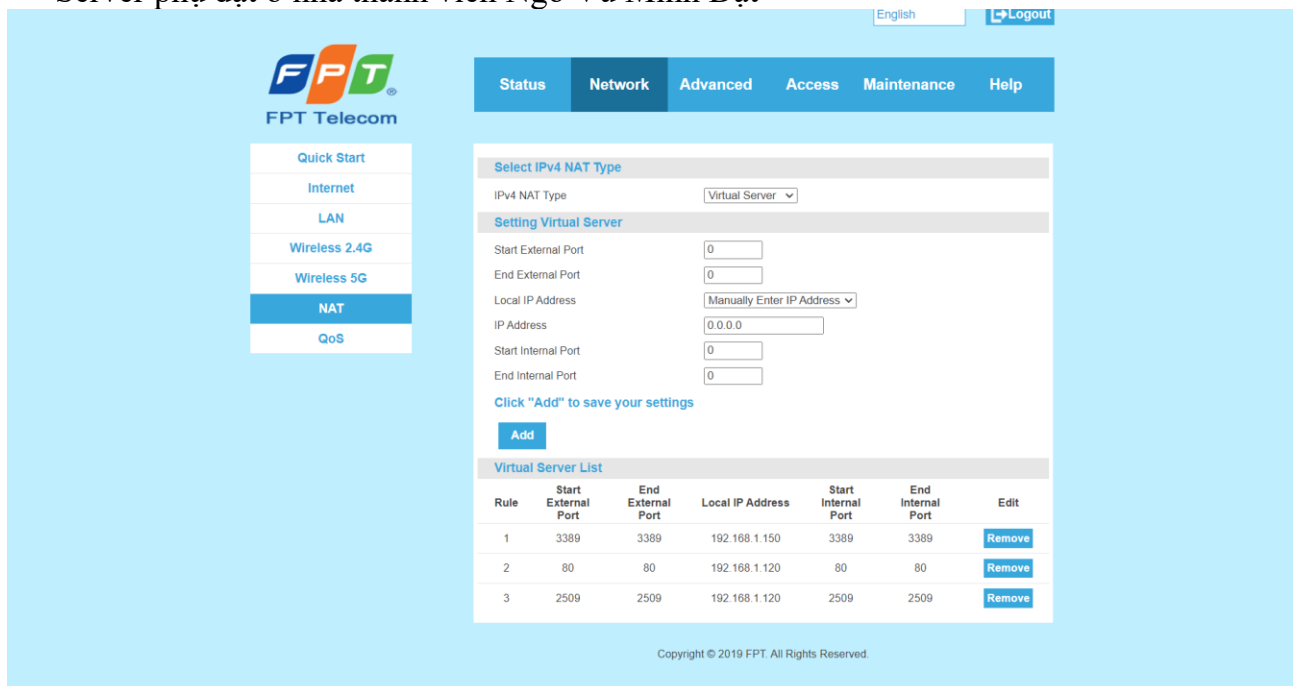
Hình 6. Inbound port rules của Vm2, nhưng lần này là cho phép port 8096 vì là JellyFin được cấu hình lắng nghe port 8096

- Trên tường lửa của Win cũng cấu hình tương tự.
- Sau khi cấu hình xong thì vm1 bật file server và vm bật Docker chạy container gồm 2 service Jellyfin và Rclone

3.5 Chức năng backup server

3.5.1 Thành phần

- Server phụ đặt ở nhà thành viên Ngô Vũ Minh Đạt



Hình 7. Cấu hình port forwarding đến server (192.168.1.120)

- Trên server chạy ứng dụng đồ án và Docker đã cấu hình sẵn

3.5.2 Nguyên lý hoạt động

- Khi main server trên Azure không may bị lỗi thì Client sẽ ngắt kết nối và client tìm địa chỉ ip phụ thông qua dns, dns name được sử dụng trong đồ án là ngovuminhdat.ddns.net

- Sau khi tìm được địa chỉ IP thì Client dùng địa chỉ này kết nối với port 2509. Modem kiểm tra gói tin sẽ forward đến 192.168.1.120 và thiết lập kết nối bình thường

3.5.3 Kết quả

- <https://www.youtube.com/watch?v=OjXG7ao1QmI>

3.6 Chức năng xem highlight

3.6.1 Thành phần

- Dữ liệu video được lưu trên Onedrive
- Jellyfin là phần mềm media server mã nguồn mở, nó sẽ đảm nhận việc streaming cho client, việc streaming sẽ dùng giao thức http
- Rclone là phần mềm mã nguồn mở có chức năng clone một storage trên cloud về máy mình và sử dụng như đang cài trên ổ cứng thật

3.6.2 Nguyên Lý hoạt động

- Khi Client request xem highlight, gói tin sẽ đến server, sau đó server sử dụng API của JellyFin để lấy thông tin về link stream của các video hiện có. Sau đó đóng gói gói tin lại và gửi cho Client. Lúc này Client chọn xem video nào sẽ có link stream tương ứng, Client dùng link stream này kết nối tới địa chỉ public tương tự server nhưng với port là 8096
- Khi JellyFin nhận request từ Client, nó sẽ giao tiếp với rclone để kéo dữ liệu từ Onedrive và gửi cho Client
- Client nhận dữ liệu và dùng trình player video mặc định của win để xem video

3.6.3 Kết quả

<https://www.youtube.com/watch?v=WRqjUfGaJA8>

4 Bảng phân chia công việc:

STT	MSSV	Tên	Nội dung công việc phụ trách	Mức độ hoàn thành	Tự đánh giá (theo thang điểm 10)
1	21521935	Ngô Vũ Minh Đạt	<ul style="list-style-type: none"> • Xây dựng class library football, SqlUser, JellyFin API, APIUser (Rapid API) • Khả năng giao tiếp giữa client và server • Khả năng giao tiếp giữa server và Rapid API • Khả năng giao tiếp giữa rclone và OneDrive • Khả năng giao tiếp giữa Server và JellyFin • Kết hợp Rclone và Jellyfin triển khai chạy trên Docker 	100%	10

			<ul style="list-style-type: none"> • Khả năng giao tiếp giữa Client và JellyFin (video streaming) • Triển khai toàn bộ hệ thống trên Azure 		
2	21522115	Lê Phi Hùng	<ul style="list-style-type: none"> • Phát triển thêm chức năng kết nối server phụ khi chức năng chính hỏng • Chịu trách nhiệm sửa lỗi mô hình Client và server còn hoạt động sai • Phát triển tính năng dấu API key và connection string SQL 	100%	9
3	21522243	Đỗ Phúc Kiên	<ul style="list-style-type: none"> • Phát triển giao diện • Phát triển tính năng đăng kí, quên mật khẩu có thông báo qua email dựa trên class SqlUser • Viết báo cáo, soạn slide Powerpoint 	100%	9
4	21521947	Nguyễn Văn Đạt	<ul style="list-style-type: none"> • Xây dựng các form trong đồ án, chịu trách nhiệm đảm bảo các class library không bug • Đọc tài liệu API Rapid và Jellyfin API để xây dựng rapid API và Jellyfin API 	100%	9

HẾT