**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

------------------------------



**BÁO CÁO THỰC TẬP**

**TỐT NGHIỆP ĐẠI HỌC**

***Đề tài*: Xây dựng ứng dụng quản lý thông tin mạng LAN của trường học**

**Người hướng dẫn : TS. NGUYỄN HỒNG SƠN**

**Sinh viên thực hiệ : NGUYỄN VĂN ANH**

**Mã số sinh viên : N19DCCN009**

**Lớp : D19CQCNHT01-N**

**Khoá** **: 2019 - 2023**

**Hệ** **: ĐẠI HỌC CHÍNH QUY**

**TP.HCM, tháng 8/2023**

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

------------------------------



**BÁO CÁO THỰC TẬP**

**TỐT NGHIỆP ĐẠI HỌC**

***Đề tài*: Xây dựng ứng dụng quản lý thông tin mạng LAN của trường học**

**Người hướng dẫn : TS. NGUYỄN HỒNG SƠN**

**Sinh viên thực hiệ : NGUYỄN VĂN ANH**

**Mã số sinh viên : N19DCCN009**

**Lớp : D19CQCNHT01-N**

**Khoá** **: 2019 - 2023**

**Hệ** **: ĐẠI HỌC CHÍNH QUY**

**TP.HCM, tháng 8/2023**

**LỜI CẢM ƠN**

Lời đầu tiên, em xin phép gửi lời tri ân sâu sắc đến các thầy cô trường Học Viện Công Nghệ Bưu Chính Viễn Thông cơ sở tại TP.HCM đã tận tình dẫn dắt và truyền đạt cho em rất nhiều kiến thức quý báu trong các học kì vừa qua.

Đặc biệt, em xin gửi lời cảm ơn chân thành tới TS.Nguyễn Hồng Sơn. Thầy đã hướng dẫn tận tình, truyền đạt kiến thức, hướng dẫn em cách nghiên cứu, đọc tài liệu trong suốt thời gian học tập và thực hiện đề tài. Kính chúc thầy và gia đình nhiều sức khỏe và thành công trong cuộc sống

Tuy nhiên, vì thời gian, kiến thức và kinh nghiệm có hạn, và cũng là lần đầu tiên được tiếp xúc, làm việc thực tế tại công ty nên bài làm của em còn có nhiều thiếu sót trong việc trình bày, đánh giá và đề xuất ý kiến. Em rất mong nhận được sự thông cảm và đóng góp ý kiến của quý thầy cô và các bạn.

Sau cùng em xin cảm ơn tất cả bạn bè, những người luôn quan tâm và ủng hộ em.

Một lần nữa, em xin chân thành cảm ơn!

TP Hồ Chí Minh, tháng 8 năm 2023

SINH VIÊN THỰC HIỆN ĐỀ TÀI

Nguyễn Văn Anh

**MỞ ĐẦU**

Trong nhiều năm gần đây hệ thống mạng máy tính là phần không thể thiếu trong các tổ chức hay các công ty, trường học. Trong điều kiện kinh tế hiện nay hầu hết đa số các tổ chức hay các công ty, trường học có phạm vi sử dụng bị giới hạn bởi diện tích và mặt bằng đều triển khai xây dựng mạng LAN để phục vụ cho việc quản lý dữ liệu nội bộ cơ quan mình được thuận lợi, đảm bảo tính an toàn dữ liệu cũng như tính bảo mật dữ liệu mặt khác mạng Lan còn giúp các nhân viên trong các tổ chức, nhà trường hay công ty truy nhập dữ liệu một cách thuận tiện với tốc độ cao. Một điểm thuận lợi nữa là mạng LAN còn giúp cho người quản trị mạng phân quyền sử dụng tài nguyên cho từng đối tượng là người dùng một cách rõ ràng và thuận tiện giúp cho những người có trách nhiệm lãnh đạo công ty, tổ chức hay nhà trường đó dễ dàng quản lý nhân viên, học sinh và điều hành công ty.

Và cùng thời gian đó thuật ngữ NoSQL (Not-only SQL) đã không còn quá xa lạ đối với chúng ta. Ra đời từ những năm 1998, NoSQL ám chỉ những cơ sở dữ liệu không dùng mô hình dữ liệu quan hệ để quản lý dữ liệu. Cơ sở dữ liệu đồ thị (Graph Database) cũng là một dạng NoSQL dựa trên lý thuyết đồ thị để mô hình hóa dữ liệu và truy vấn. Câu hỏi đặt ra là tại sao lại có sự xuất hiện của các loại cơ sở dữ liệu trên. Ta có thể thấy hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) vẫn được dùng một cách phổ biến trong thời điểm này. Nhưng với lượng dữ liệu ngày càng lớn, mức độ tăng nhanh khiến cho việc xử lý nhiều truy vấn phức tạp rất mất thời gian và hay gây lỗi. Chưa kể tới cơ sở dữ liệu quan hệ có cấu trúc rất chặt chẽ, khó thay đổi, khó mở rộng và nâng cấp… Có nhiều nguyên nhân và nhiều yếu tố dẫn đến sự ra đời của NoSQL và cơ sở dữ liệu đồ thị. Và tất nhiên không phải ứng dụng nào cũng cần tới NoSQL hay cơ sở dữ liệu đồ thị. Tất cả đều mang tính tương đối và cân nhắc hợp lý.

Trong tài liệu này,tôi sẽ trình bày cơ sở dữ liệu đồ thị là gì, có những ưu nhược điểm và tính chất gì để từ đó biết được nên áp dụng chúng vào những ứng dụng như thế nào. Sự khác biệt giữa cơ sở dữ liệu đồ thị và cơ sở dữ liệu quan hệ là gì? Ngôn ngữ truy vấn trong cơ sở dữ liệu đồ thị khác ngôn ngữ truy vấn trong cơ sở dữ liệu quan hệ ra sao. Một phần nữa của tài liệu sẽ là phần áp dụng cơ sở dữ liệu đồ thị vào một ứng dụng cụ thể, đó là : Ứng dụng quản lý thông tin mạng LAN của trường học. Trong đó cũng nêu rõ mục đích vì sao ứng dụng này lại cần tới cơ sở dữ liệu đồ thị và nó sử dụng cơ sở dữ liệu đồ thị để mô hình hóa dữ liệu một cách cụ thể thế nào.

**Mục lục**

[CHƯƠNG 1: TỔNG QUAN VỀ MẠNG MÁY TÍNH 6](#_Toc142830633)

[1.1 Khái niệm mạng máy tính 6](#_Toc142830634)

[1.2 Phân loại mạng máy tính 7](#_Toc142830635)

[1.3 Mạng LAN và thiết kế mạng LAN 8](#_Toc142830636)

[CHƯƠNG 2: CƠ SỞ DỮ LIỆU ĐỒ THỊ 10](#_Toc142830637)

[2.1 Nhược điểm của cơ sở dữ liệu khác 10](#_Toc142830638)

[2.2 Đồ thị là gì ? Đặc điểm của đồ thị 11](#_Toc142830639)

[2.3 Cơ sở dữ liệu đồ thị 12](#_Toc142830640)

[2.4 Ngôn ngữ xử lý dữ liệu trong CSDL đồ thị Neo4j 15](#_Toc142830641)

[Chương 3: Giới thiệu công cụ, thư viện sử dụng trong đồ án và các cơ sở dữ liệu đồ thị 18](#_Toc142830642)

[3.1 Pycharm 18](#_Toc142830643)

[3.2 Tkinter 19](#_Toc142830644)

[3.3 Một số cơ sở dữ liệu đồ thị 20](#_Toc142830645)

[**3.3.1 Neo4j:** 20](#_Toc142830646)

[**3.3.2 Amazon Neptune:** 21](#_Toc142830647)

[**3.3.3 Microsoft Azure Cosmos DB:** 21](#_Toc142830648)

[**3.3.4 OrientDB: OrientDB** 21](#_Toc142830649)

[Chương 4: Yêu cầu đặt ra và giải quyết yêu cầu 22](#_Toc142830650)

[4.1 Khảo sát trường 22](#_Toc142830651)

[4.2 Yêu cầu hệ thống 23](#_Toc142830652)

[4.3 Yêu cầu thiết kế 23](#_Toc142830653)

[Chương 5: Triển khai thiết kế 24](#_Toc142830654)

[5.1 Sơ đồ logic 24](#_Toc142830655)

[5.2 Thiết kế csdl 27](#_Toc142830656)

[Chương 6: Demo ứng dụng đã xây dựng 33](#_Toc142830657)

[Kết luận và đánh giá 41](#_Toc142830658)

[DANH MỤC TÀI LIỆU THAM KHẢO 42](#_Toc142830659)

**Mục lục hình ảnh**

[Hình 1. mô tả mạng máy tính 6](file:////Users/tony/Documents/NguyenHongSon_D19CQCNHT01-N_NguyenVanAnh_BCCK/NguyenHongSon_D19CQCNHT01-N_NguyenVanAnh_BCCK.docx#_Toc142830120)

[Hình 2 mô tả mạng LAN 7](file:////Users/tony/Documents/NguyenHongSon_D19CQCNHT01-N_NguyenVanAnh_BCCK/NguyenHongSon_D19CQCNHT01-N_NguyenVanAnh_BCCK.docx#_Toc142830121)

[Hình 3 mô tả mạng WAN 8](file:////Users/tony/Documents/NguyenHongSon_D19CQCNHT01-N_NguyenVanAnh_BCCK/NguyenHongSon_D19CQCNHT01-N_NguyenVanAnh_BCCK.docx#_Toc142830122)

[Hình 4 mô hình CSDL quan hệ 10](#_Toc142830123)

[Hình 5 ví dụ về đồ thị 11](#_Toc142830124)

[Hình 6 ví dụ đồ thị thuộc tính 12](#_Toc142830125)

[Hình 7 đồ thị cơ sở dữ liệu 13](#_Toc142830126)

[Hình 8 kết quả khi kiểm tra tốc độ giữa các csdl 14](#_Toc142830127)

[Hình 9 mô hình ví dụ ngôn ngữ dữ liệu trong csdl 15](#_Toc142830128)

[Hình 10 sơ đồ logic thiết kế mạng 24](#_Toc142830129)

[Hình 11 Sơ đồ logic của Switch Layer 1 24](#_Toc142830130)

[Hình 12 Sơ đồ logic của Switch Laye 3 ……………………………………….. Hình 13 Sơ đồ logic của Switch Layer 2 25](#_Toc142830131)

[Hình 14 Bảng subet được chia theo sơ đồ 26](#_Toc142830132)

[Hình 15 Dữ liệu ở switch 1 của switch layer 1 27](#_Toc142830133)

[Hình 16 Dữ liệu ở Switch 3 của Switch layer 1 27](#_Toc142830134)

[Hình 17 Dữ liệu ở Switch 8 của Switch layer 2 28](#_Toc142830135)

[Hình 18 Dữ liệu ở Switch 13 của Switch layer 3 28](#_Toc142830136)

[Hình 19 Bảng dữ liệu node switch layer 29](#_Toc142830137)

[Hình 20 Bảng dữ liệu node switch 29](#_Toc142830138)

[Hình 21Bảng dữ liệu node server 30](#_Toc142830139)

[Hình 22 Bảng dữ liệu node router 30](#_Toc142830140)

[Hình 23 Bảng dữ liệu node room 31](#_Toc142830141)

[Hình 24 Bảng dữ liệu node PC 31](#_Toc142830142)

[Hình 25 dữ liệ trong hệ thống 32](#_Toc142830143)

[Hình 26 các ràng buộc trong csdl 32](#_Toc142830144)

[Hình 27 Giao diện đăng nhập 33](#_Toc142830145)

[Hình 28 Giao diện đăng ký: (quyền mặc định là User) 33](#_Toc142830146)

[Hình 29 Giao diện Chính Khi load Thành Công PC 34](#_Toc142830147)

[Hình 30 Giao diện khi chuyển chế độ sáng 34](#_Toc142830148)

[Hình 31Giao diện đổi mật khẩu 35](#_Toc142830149)

[Hình 32 Giao diện Tạo Tài Khoản Admin 35](#_Toc142830150)

[Hình 33 File mật khẩu username và quyền: 36](#_Toc142830151)

[Hình 34 Thanh phụ hiển thị thông tin Phòng 36](#_Toc142830152)

[Hình 35 Thanh phụ hiển thị ip range 36](#_Toc142830153)

[Hình 36 Thanh Phụ hiển thị Maphong khả dụng, Ip khả dụng và Network khả dụng 36](#_Toc142830154)

[Hình 37 Giao diện khi edit PC 36](#_Toc142830155)

[Hình 38 Tính năng thêm PC vào switch 37](#_Toc142830156)

[Hình 39 Cơ sở dữ liệu sau khi thêm PC có mã TV 6 37](#_Toc142830157)

[Hình 40 Giao diện khi xóa PC 38](#_Toc142830158)

[Hình 41 Cơ sở dữ liệu sau khi xóa PC 38](#_Toc142830159)

[Hình 42 Giao diện admin switch có thể thêm xóa sửa switch 39](#_Toc142830160)

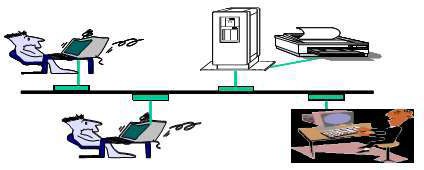
[Hình 43 Giao diện User PC có thể xem thông tin PC 39](#_Toc142830161)

[Hình 44 Giao diện User Switch có thể xem thông tin switch 40](#_Toc142830162)

# **CHƯƠNG 1: TỔNG QUAN VỀ MẠNG MÁY TÍNH**

## 1.1 Khái niệm mạng máy tính

Về cơ bản, mạng máy tính là hai hay nhiều máy tính được kết nối với nhau theo một cách nào đó. Khác với các trạm truyền hình gửi thông tin đi, chỉ có người nhận thông tin, không thể phản hồi lại thông tin đó, mạng máy tính luôn hai chiều, khi máy tính A gủi thông tin tới máy tính B thì B có thể trả lời lại A.

Nói cách khác, một số máy tính được kết nối với nhau và có thể trao đổi thông tin cho nhau gọi là mạng máy tính.

Hình . mô tả mạng máy tính

Mạng máy tính ra đời xuất phát từ nhu cầu muốn chia sẻ và dùng chung dữ liệu. Nếu không có mạng máy tính thì dữ liệu muốn chia sẻ trên các máy tính độc lập với nhau phải thông qua việc in ấn, sao chép qua usb, cd-rom… điều này gây bất tiện rất lớn đối với người dùng. Để giải quyết khó khăn đó, kết nối những máy tính độc lập lại thành mạng máy tính thì chúng ta sẽ có những ưu điểm sau:

* Nhiều người có thể dùng chung một phần mềm tiện ích.
* Một nhóm người cùng thực hiện một công việc thì họ sẽ dùng chung dữ liệu, dùng chung tập tin của công việc đó,trao đổi thông tin với nhau dễ dàng hơn.
* Dữ liệu được quản lý tập trung nên an toàn hơn, trao đổi với nhau nhanh chóng, thuận lợn hơn.
* Có thể dùng chung các thiết bị ngoại vi đắt tiền (máy in, máy vẽ, máy photocopy,…)
* Người dùng trao đổi với nhau qua hộp thư điện tử dễ dàng (Email) và có thể sử dụng như là một công cụ để thông báo tin tức, nội dung của một buổi hợp, thời khóa biểu…
* An toàn cho dữ liệu và phần mềm vì phần mềm mạng sẽ khóa các tệp khi có những người dùng không đủ quyền truy xuất các tập tin và thư mục đó.

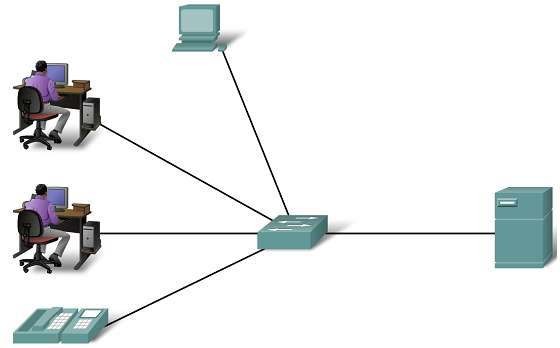
Một hệ thống mạng tổng quát được cấu tạo từ 3 thành phần:

* Đường biên mạng (Network Edge): gồm các máy tính (host) và các chương trình ứng dụng mạng (Network application).
* Đường trục mạng (Network Core): gồm các bộ chọn đường (router) đóng vai trò là một mạng trung tâm kết nối các mạng lại với nhau.
* Đường truyền vật lý (Physical media): gồm các đường truyền tải thông tin.

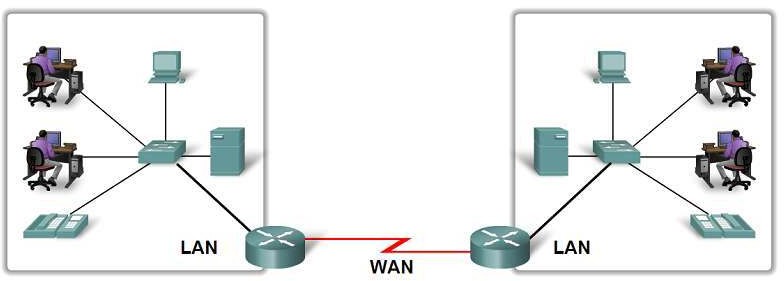
## 1.2 Phân loại mạng máy tính

Mạng máy tính có thể phân bố trên vùng lãnh thổ nhất định và có thể phân bố trong phạm vi một quốc gia hay quốc tế.

Dựa vào đường kính của mạng người ta có thể phân ra các loại mạng như sau:

* Mạng cục bộ LAN (Local Area Network) : là mạng được thiết kế trong phạm vi nhỏ khoảng cách giữa các nút mạng nhỏ hơn 10km. LAN thường được sử dụng trong phạm vi nội bộ cơ quan, xí nghiệp… Lan là một loại mạng quảng bá, sử dụng chung một đường truyền chia sẻ, có sự canh tranh đường truyền. Các LAN được kết nối với nhau thành một WAN (mạng diện rộng).

Hình mô tả mạng LAN

* Mạng đô thị MAN (Metropolitan Area Network) : là mạng được thiết kế trong phạm vi đô thị, trong một tỉnh thành có bán kính khoảng 100km trở lại. Mạng diện rộng WAN (Wide Area Network) : phạm vi của mạng vượt qua biên giới quốc gia và thậm chí là cả châu lục.

Hình mô tả mạng WAN

* Mạng toàn cầu Gan (Global Area Network) : là mạng được thiết lập trên phạm vi toàn cầu, khắp các châu lục trên trái đất. Thông thường kết nối thông qua mạng viễn thông và vệ tinh.

Trong các khái niệm trên, WAN và LAN là hai khái niệm được sử dụng nhiều nhất.

**1.3 Mạng LAN và thiết kế mạng LAN**

* Mạng cục bộ LAN là thống truyền thông tốc độ cao được thiết kế để kết nối các máy tính và các thiết bị khác nhau cùng hoạt động với nhau trong một khu vực địa lý nhỏ như một tầng của tòa nhà, hoặc một tòa nhà, một trường học… Một số mạng Lan có thể kết nối lại với nhau trong một khu làm việc. Các mạng Lan trở nên thông dụng vì nó cho phép những người sử dụng dùng chung những tài liệu quan trọng hoặc một số thiết bị đắt tiền như máy in, ổ đĩa DVD – ROM, các phần mềm ứng dụng và những thông tin cần thiết khác.

Các thiết bị kết nối mạng LAN:

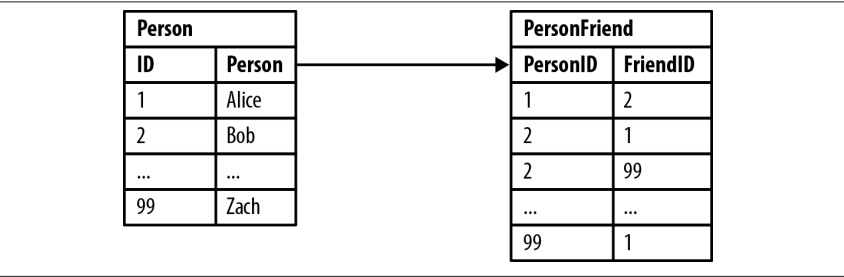
* Card mạng – NIC (Network Interface Card): là một thiết bị được cắm vào trong máy tính để cung cấp cổng kết nối vào mạng. Card mạng được coi là một thiết bị hoạt động ở lớp 2 của mình OSI. Mỗi card mạng có chứa một địa chỉ duy nhất là địa chỉ MAC – Media Access Control. Card mạng điều khiển việc kết nối của máy tính vào các phương tiện truyền dẫn trên mạng.
* Repeater: là một thiết bị hoạt động ở tầng 1 của mô hình OSI, khuếch đại và định hình lại tín hiệu. Repeater gửi mọi tín hiệu mà nó nhận được từ một port ra tất cả các port còn lại. Chức năng của repeater là phục hồi lại các tín hiệu trên đường truyền mà không sửa đổi gì.
* Hub được coi là một Repeater có nhiều cổng. Một Hub có từ 4 đến 24 cổng và có thể còn nhiều hơn. Trong phần lớn các trường hợp, Hub được sử dụng trong các mạng 10BASE-T hay 100BASE-T. Khi cấu hình mạng là hình sao (Star topology), Hub đóng vai trò là trung tâm của mạng. Với một Hub, khi thông tin vào từ một cổng và sẽ được đưa đến tất cả các cổng khác.
* Bridge: là thiết bị mạng thuộc lớp 2 của mô hình OSI (Data Link Layer). Bridge được sử dụng để ghép nối 2 mạng để tạo thành một mạng lớn duy nhất. Bridge được sử dụng phổ biến để làm cầu nối giữa hai mạng Ethernet. Bridge quan sát các gói tin (packet) trên mọi mạng. Khi thấy một gói tin từ một máy tính thuộc mạng này chuyển tới một máy tính trên mạng khác, Bridge sẽ sao chép và gửi gói tin này tới mạng đích.
* Router: là thiết bị mạng lớp 3 của mô hình OSI (Network Layer). Router kết nối hai hay nhiều mạng IP với nhau. Các máy tính trên mạng phải “nhận thức” được sự tham gia của một router, nhưng đối với các mạng IP thì một trong những quy tắc của IP là mọi máy tính kết nối mạng đều có thể giao tiếp được với router.
* Switch đôi khi được mô tả như là một Bridge có nhiều cổng. Trong khi một Bridge chỉ có 2 cổng để liên kết được 2 segment mạng với nhau, thì Switch lại có khả năng kết nối được nhiều segment lại với nhau tuỳ thuộc vào số cổng (port) trên Switch. Cũng giống như Bridge, Switch cũng “học” thông tin của mạng thông qua các gói tin (packet) mà nó nhận được từ các máy trong mạng. Switch sử dụng các gói thông tin đến đúng địa chỉ. Ngày nay, trong các giao tiếp dữ liệu, Switch thường có 2 chức năng chính là chuyển các khung dữ liệu từ nguồn đến đích, và xây dựng các bảng Switch. Switch thông tin này để xây dựng lên bảng Switch, bảng này cung cấp thông tin giúp các hoạt động ở tốc độ cao hơn nhiều so với Repeater và có thể cung cấp nhiều chức năng hơn như khả năng tạo mạng LAN ảo (VLAN)

# **CHƯƠNG 2: CƠ SỞ DỮ LIỆU ĐỒ THỊ**

## 2.1 Nhược điểm của cơ sở dữ liệu khác

CSDL quan hệ được xây dựng với các bảng, trường, bản ghi và có mối quan hệ với nhau thông qua các khóa chính, khóa ngoại.. Mặc dù nói là có quan hệ với nhau thông qua các phép join (nối bảng) nhưng như thế vẫn chưa đáp ứng được tính chất của các dữ liệu kết nối. Làm thế nào để đưa các dữ liệu kết nối và các dữ liệu nửa cấu trúc vào CSDL quan hệ - CSDL lưu trữ dữ liệu theo một định dạng cố định và trong một hệ thống các bảng có cấu trúc. Lấy một cái có cấu trúc để lưu trữ cái nửa cấu trúc thực sự là một vấn đề nan giải. Tệ hơn nữa khi cấu trúc của bộ dữ liệu ngày càng phức tạp và không đồng đều bởi các mối quan hệ, sự gia tăng trong việc liên kết các giao dịch ngày càng lớn khiến cho CSDL quan hệ không còn đáp ứng được việc xử lý, truy vấn và lưu trữ dữ liệu. Điều này dẫn đến sự ra đời của loại CSDL khác phù hợp hơn.

Có thể ví dụ về mô hình hóa dữ liệu kết nối trong CSDL quan hệ như dưới đây:



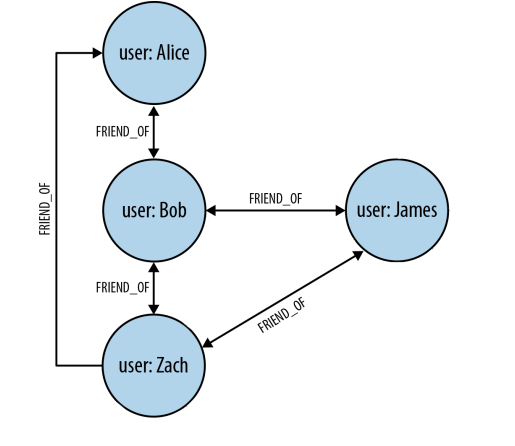
Hình mô hình CSDL quan hệ

Có thể xét ví dụ trên để thấy được sự thiếu tính quan hệ của các csdl NoSQL. Ở đây ta sử dụng các lưu trữ liên kết. Tức là với phần lưu trữ user : Alice thì ta có các bảng liên kết phụ khác như order và item. Điều này dễ dàng khiến ta tưởng rằng có thể dùng cặp khóa-giá trị để quản lý đồ thị. Nhưng thực tế thì không dùng được các cặp khóa- giá trị như thế vì đây chỉ là bảng phụ (hỗ trợ, tham khảo) và là sự liên kết cả bảng chứ không theo một cặp giá trị nào. Chưa kể khi ta cập nhật bảng user : Alice thì cũng phải nhớ cập nhật các bảng liên kết phụ, nếu không thông tin sẽ bị xung đột và không bảo toàn. Còn nếu phải xóa bảng lưu trữ chính đi thì nếu bảng liên kết phụ không được xóa cũng sẽ gây ra lãng phí không gian cũng như dư thừa dữ liệu. Như vậy việc lưu trữ dữ liệu kết nối với NoSQL không phải lúc nào cũng thuận tiện và mang lại hiệu quả.

## 2.2 Đồ thị là gì ? Đặc điểm của đồ thị

Thông thường đồ thị được định nghĩa là một tập các đỉnh được nối với nhau bởi các cạnh, hay có thể nói là gồm các nút và các mối quan hệ giữa các nút với nhau. Nếu coi các nút là các thực thể, và chúng được nối với nhau bởi các mối quan hệ thì đồ thị có thể mô hình hóa bất kỳ đối tượng nào, vấn đề nào, tình huống nào trong thế giới thực. Từ việc mô hình một cái tủ đựng đồ cho tới cái máy bay hay mô hình các quan hệ trên mạng xã hội, mô hình sự đa dạng của sinh vật sống, mô hình một quy trình sản xuất, mô hình một cuộc khởi nghĩa…

Bởi vì thế giới thực quá đa dạng và có mối quan hệ với nhau chứ không tuân theo một tập quy luật bất biến nào nên muốn lưu trữ hay xử lý các thông tin như thế này được hiệu quả, chúng ta nên dùng tới đồ thị. Và thực tế thì người ta đã ứng dụng đồ thị vào rất nhiều vấn đề trong cuộc sống. Có thể lấy ví dụ đơn giản trên mạng xã hội facebook như sau :



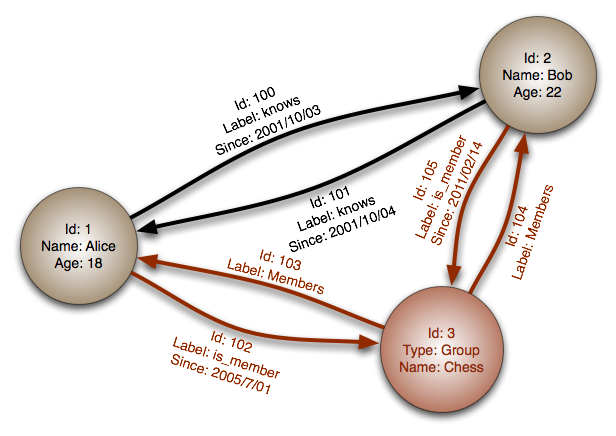
Hình ví dụ về đồ thị

Trong đồ thị trên, các nút biểu diễn các thực thể là các người dùng được định danh bằng tên và có các mối quan hệ là FRIEND\_OF. Mối quan hệ trong đồ thị có thể là một chiều, có thể là hai chiều. Như trong đồ thị trên thì hầu hết là mối quan hệ hai chiều, chỉ có mối quan hệ giữa Zach và Alice là một chiều. Như vậy đồ thị trên hoàn toàn có thể biểu diễn các mối quan hệ trong đời thực một cách dễ dàng và trực quan.

Mỗi thực thể (đối tượng) đều có những đặc điểm riêng, có thể hoàn toàn đưa các đặc điểm đó thành các nút, nhưng như vậy sẽ rất phức tạp, nhất là khi số thực thể tăng lên và các nút đặc điểm đó lại chỉ nối với một thực thể. Như vậy sẽ gây lãng phí số nút, không gian và làm phức tạp đồ thị một cách không cần thiết. Các mô hình đồ thị thuộc tính đã giải quyết được vấn đề này. Trong các mô hình đồ thị thuộc tính có các đặc điểm sau :

* Bao gồm các nút và các mối quan hệ
* Các nút có các thuộc tính (thuộc tính thể hiện theo cặp : khóa – giá trị)
* Các mối quan hệ được đặt tên và được đặt trực tiếp vào đồ thị, luôn có nút đầu và nút cuối
* Các mối quan hệ cũng có thể bao gồm các thuộc tính

Ví dụ về đồ thị thuộc tính như sau :



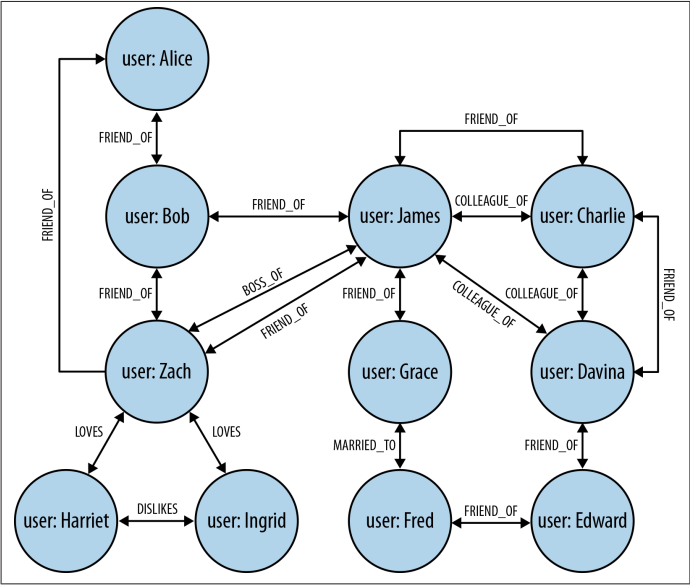
Hình ví dụ đồ thị thuộc tính

Trong đồ thị trên có 3 nút được phân biệt bởi Id và có các thuộc tính Name và Age với các cặp khóa-giá trị như hình vẽ. Như nút có Id=1 thì có thuộc tính Name = Alice và Age = 18. Tương tự 2 nút kia cũng có thuộc tính như vậy. Ngoài ra thì mối quan hệ cũng hoàn toàn có thể có thuộc tính. Ví dụ mối quan hệ từ nút 1 đến nút 2 có Id = 100, thuộc tính là Label = knows và Since=2001/10/03. Tương tự các mối quan hệ giữa các nút khác cũng như vậy. Thuộc tính ở mối quan hệ hay ở nút đều tồn tại theo cặp khóa-giá trị.

**2.3 Cơ sở dữ liệu đồ thị**

Với vai trò như người dùng, chúng ta có thể hoàn toàn suy luận ra các phụ thuộc ngữ nghĩa giữa các thực thể, tuy nhiên các mô hình dữ liệu và csdl của chúng thì không được hiện ra. Và để đáp ứng được điều này, các ứng dụng phải tạo một mạng riêng, không kết nối với dữ liệu ngay tại ứng dụng, sau đó giải quyết các truy vấn chậm và tiềm ẩn có thể phát sinh.

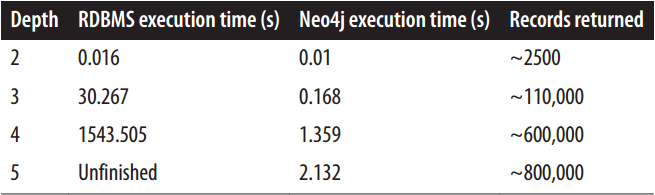
Trong thế giới đồ thị, lưu trữ dữ liệu kết nối chính là kết nối dữ liệu. Ở đâu có các kết nối trong một phạm vi chủ đề thì có kết nối trong dữ liệu. Cùng xem xét ví dụ dưới đây để thấy rõ hơn.



Hình đồ thị cơ sở dữ liệu

Trong sơ đồ trên có thể thấy có rất nhiều trường hợp thực tế của dữ liệu có kết nối, các kết nối giữa các thực thể không thể hiện được tính không đồng nhất trong một phạm vi nhất định (thể hiện tính nửa cấu trúc). Và với một mạng đơn giản thế này, chúng ta cũng thấy được việc thêm bớt các nút để tăng kích thước mạng là hoàn toàn có thể và linh động, không ảnh hưởng tới các nút khác cũng như các mối quan hệ giữa các nút trong mạng. Như vậy cũng không cần lo lắng tới việc di chuyển hay thay đổi cấu trúc dữ liệu trong mạng, đồng thời vẫn giữ được dữ liệu và các quan hệ gốc.

Mô hình hóa dữ liệu như trên cho ta dễ dàng thấy được các mối quan hệ của một thực thể với các thực thể còn lại thông qua các hướng của các mối quan hệ. Có thể thấy như là Zach là bạn của Alice nhưng chiều ngược lại thì không – điều này nếu thực hiện trong CSDL quan hệ như xét ở trên thì khá lằng nhằng và phức tạp. Và những câu truy vấn như : bạn của bạn của bạn.. thì sử dụng CSDL đồ thị sẽ dễ dàng hơn rất nhiều so với việc sử dụng các CSDL khác. Nhất là khi độ sâu của truy vấn tăng lên. Điều này đã được Vukotic và bạn đồng hành thử nghiệm trên một mạng xã hội với độ sâu tối đa là 5 [Graph Databases – 2013]. Kết quả thể hiện ở bảng sau :



Hình kết quả khi kiểm tra tốc độ giữa các csdl

Bảng này cho biết thời gian thực hiện truy vấn và số lượng bản ghi trả về của một

hệ quản trị csdl quan hệ và một hệ quản trị csdl đồ thị (Neo4j).

Có thể thấy sự chênh nhau đáng kể của hệ QT CSDL quan hệ qua các lần truy vấn và cho tới mức độ sâu bằng 5 thì nó đã không còn hoàn thành được truy vấn. Trong khi đó với Neo4j thì vẫn thực hiện truy vấn ổn định và thời gian thì nhanh hơn hẳn csdl quan hệ (độ sâu càng cao càng thấy rõ sự khác biệt). Như vậy rõ ràng là csdl đồ thị hiệu quả hơn hẳn csdl quan hệ khi làm việc với dữ liệu kết nối. Tuy nhiên như vậy không có nghĩa là csdl quan hệ và các csdl NoSQL khác không hữu ích và hiệu quả. Chúng vẫn làm việc tốt với các dữ liệu không có tính kết nối cao và ổn định, không phải thay đổi nhiều. Tùy theo tính chất của từng loại dữ liệu, người ta sẽ xem xét nên sử dụng loại csdl nào cho phù hợp.

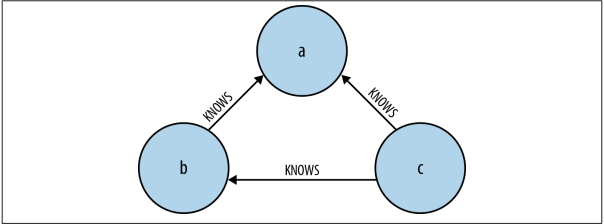
Từ ví dụ mạng xã hội ở trên, chúng ta hoàn toàn có thể áp dụng vào các vấn đề khác với phạm vi lĩnh vực khác trong đời sống. Quản lý thông tin sinh học, bản đồ gen, phác đồ điều trị dựa trên bệnh án của bệnh nhân và gia đình, hay gợi ý sản phẩm trong kinh doanh dựa vào lịch sử mua hàng.. tất cả đều có thể áp dụng csdl đồ thị để có được các quyết định chính xác hơn. Có thể tìm hiểu vấn đề này rõ hơn với khai phá dữ liệu và phân tích dữ liệu trực tuyến. Từ đó có thể thấy tiềm năng của csdl đồ thị là rất lớn. Tuy nhiên, báo cáo này sẽ đi sâu hơn về việc làm thế nào để xây dựng một ứng dụng sử dụng csdl đồ thị với neo4j nên tiềm năng của csdl đồ thị xin dừng ở đây. Các chương tiếp theo sẽ đi sâu về cách mô hình hóa và xây dựng csdl đồ thị cũng như áp dụng cụ thể xây dựng một ứng dụng thực.

Một điểm khác biệt rõ ràng nữa giữa csdl quan hệ và csdl đồ thị chính là ngôn ngữ xử lý dữ liệu. Với csdl đồ thị, nó chính là các phương pháp xử lý đồ thị. Theo lý thuyết đồ thị thì ta có thể thấy các loại giải thuật đồ thị gồm có : vẽ đồ thị, vẽ lại đồ thị, mở rộng cây, các luồng mạng lưới, các bài toán tìm đường đi của người bán hàng. Đây là những loại chung nhất của các giải thuật về đồ thị. Khi chia nhỏ ra thì ta sẽ thấy những giải thuật quen thuộc hơn như A\*, Floyd, Kruskal, Dijkstra.. Và thực tế trong bộ xử lý dữ liệu của csdl đồ thị có bao gồm các giải thuật về đồ thị để hỗ trợ xử lý được nhanh và hiệu quả nhất có thể. Điều này cho thấy sử dụng csdl đồ thị với dữ liệu được mô hình hướng đồ thị sẽ có nhiều sự hỗ trợ hơn trong việc xử lý dữ liệu

**2.4 Ngôn ngữ xử lý dữ liệu trong CSDL đồ thị Neo4j**

Cypher là ngôn ngữ truy vấn cho csdl đồ thị, có đặc điểm dễ đọc và dễ hiểu đối với cả các nhà phát triển, các chuyên gia csdl và các nhà kinh doanh.

Cypher cho phép người dùng tìm kiếm thông tin trên csdl theo một mô hình cụ thể nào đó.



Hình mô hình ví dụ ngôn ngữ dữ liệu trong csdl

Trong ASCII tương đương với đồ thị trong Cypher, ta biểu diễn mối quan hệ trên như sau : (a)-[:KNOWS]->(b)-[:KNOWS]->(c), (a)-[:KNOWS]->(c)

Tuy nhiên cũng như hầu hết các ngôn ngữ khác, Cypher cũng có các câu lệnh riêng. Câu truy vấn đơn giản nhất bao gồm một mệnh đề START theo sau bởi MATCH và RETURN. Sau đây là một ví dụ đơn giản cho câu truy vấn sử dụng 3 mệnh đề trên để tìm bạn chung với người dùng có tên Michael.

START a=node:user(name= ‘Michael’)

MATCH (a)-[:KNOWS]->b –[:KNOWS]->(c), (a)-[:KNOWS]->(c)

RETURN b,c

Chúng ta sẽ tìm hiểu các câu lệnh kỹ hơn.

**START**

START mô tả một hoặc nhiều điểm bắt đầu – có thể là các nút hoặc các mối quan hệ - trong đồ thị. Những điểm bắt đầu sẽ được chọn thông qua tìm kiếm chỉ mục hoặc tìm kiếm trực tiếp dựa trên các nút và các mối quan hệ. Như ví dụ phía trên, điểm bắt đầu sẽ là điểm mà được đánh chỉ mục là user và có thuộc tính (khi đánh chỉ mục tạo nên) name có giá trị là Michael. Giá trị trả về từ việc tìm kiếm điểm này là điểm a (định danh a là do người dùng tự đặt). Định danh này được sử dụng từ đầu đến cuối truy vấn.

**MATCH**

Đây là phần đặc tả bằng ví dụ từng phần. Ta sử dụng các kí tự ASCII để đại diện cho các nút và các mối quan hệ, “vẽ” ra các dữ liệu mà chúng ta quan tâm.

Chúng ta sử dụng

* Dấu ngoặc đơn để vẽ các nút, như là : (a)
* Cặp dấu gạch ngang và lớn hơn/ nhỏ hơn để thể hiện mối quan hệ (có hướng từ gốc tới ngọn hoặc theo cả hai hướng), như là : -> và <-
* Giữa các dấu gạch ngang là các dấu ngoặc vuông và dấu hai chấm là tên các mối quan hệ, như là : [:KNOWS]

Câu ví dụ trên miêu tả một đường dẫn gồm 3 nút, một trong số đó đã được nối với định danh a, những cái còn lại nối với c. Các nút này được được kết nối bởi đường liên hệ “KNOWS”. Mô hình này về lý thuyết có thể được sử dụng nhiều lần trong dữ liệu đồ thị nên chúng ta nên ghim một phần chúng lại trong đồ thị.

Vừa rồi chúng ta đã thực hiện xong việc tìm kiếm một nút thực sự trên đồ thị với mệnh đề START – nút đại diện cho Michael. Chúng ta gắn nút này vào định danh a, rồi đưa a sang mệnh đề MATCH. Từ đó ta đã ghim mô hình này đến 1 điểm trong đồ thị. Cypher khớp các phần còn lại của mô hình vào đồ thị vào xung quanh các điểm ghim. Từ đó thì nó tìm thấy các nút khác để gắn với các định danh phù hợp. Và như vậy, định danh a đại điện cho Michael, còn các định danh b, c sẽ theo định danh a mà gắn vào các nút xung quanh khớp với điều kiện trong mệnh đề MATCH.

**RETURN**

Mệnh đề này quy định việc các nút, các mối quan hệ và các thuộc tính trong dữ liệu được nối nên được trả về như thế nào cho client. Trong ví dụ trên thì chúng ta mong muốn việc trả về là các nút kết nối với định danh b và c.

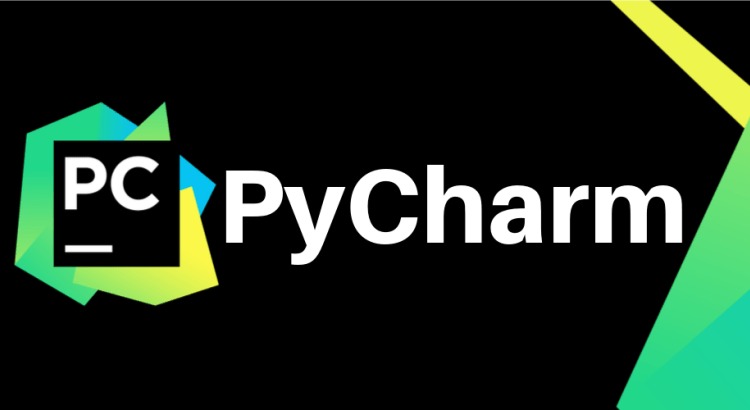
**Các câu lệnh Cypher khác:**

* WHERE : cung cấp các tiêu chí để lọc các mẫu kết quả phù hợp
* CREATE and CREAT UNIQUE : tạo nút và các mối quan hệ
* DELETE : xóa nút, mối quan hệ và thuộc tính
* SET : thiết lập các giá trị thuộc tính
* FOREACH : biểu diễn một hành động cập nhật đối với mỗi phần tử trong một danh sách
* UNION : hợp các kết quả từ 1 hoặc nhiều truy vấn
* WITH : tạo chuỗi các phần truy vấn và chuyển tiếp các kết quả từ một tới kết quả tiếp theo. Giống như tạo đường ống các lệnh trong Unix.

Cũng giống các ngôn ngữ truy vấn khác, Cypher còn nhiều mệnh đề khác như SKIP, USING, MERGE, REMOVE... Trên đây chỉ mang tính giới thiệu, chúng ta có thể tìm hiểu thêm và thực hành nhiều hơn để hiểu rõ cách dùng của các mệnh đề trên

# **Chương 3: Giới thiệu công cụ, thư viện sử dụng trong đồ án và các cơ sở dữ liệu đồ thị**

## 3.1 Pycharm



PyCharm là một môi trường phát triển toàn diện và mạnh mẽ cho ngôn ngữ lập trình Python. Được xây dựng bởi JetBrains - một công ty nổi tiếng về phát triển phần mềm - PyCharm cung cấp các công cụ hàng đầu để tạo, sửa đổi và quản lý mã nguồn Python một cách hiệu quả. Với tính năng gợi ý mã thông minh, trình gỡ lỗi tiện ích, hỗ trợ kiểm thử tự động, và tích hợp mạnh mẽ với các khung làm việc và công cụ phát triển phổ biến, PyCharm là người bạn đồng hành tốt nhất cho nhà phát triển Python.

**Tính năng nổi bật:**

* **Trình soạn thảo mã thông minh:** Tự động hoàn thành mã, kiểm tra cú pháp, và gợi ý mã dựa trên ngữ cảnh giúp bạn viết mã nhanh chóng và chính xác hơn.
* **Trình gỡ lỗi mạnh mẽ:** Điều tra mã bằng trình gỡ lỗi tích hợp, theo dõi biến và thậm chí xem trình đoạn biểu đồ (stack trace) để xác định và khắc phục lỗi.
* **Quản lý dự án tốt hơn:** PyCharm giúp bạn quản lý các tệp, thư mục và phiên bản của dự án một cách dễ dàng, bằng cách tích hợp với các hệ thống quản lý phiên bản như Git.
* **Kiểm thử tự động:** Tự động hóa quá trình kiểm thử bằng cách viết và chạy các bài kiểm tra tự động, giúp đảm bảo tính ổn định và đáng tin cậy của mã.
* **Tích hợp khung làm việc và công cụ:** PyCharm hỗ trợ nhiều khung làm việc và công cụ phát triển phổ biến như Tkinter, Django, Flask và Git.

## 3.2 Tkinter



Tkinter là một thư viện giao diện người dùng (GUI) mạnh mẽ và dễ sử dụng được tích hợp sẵn trong ngôn ngữ lập trình Python. Được xây dựng trên nền tảng toolkit Tk, Tkinter cung cấp các công cụ và tài liệu cho việc tạo các ứng dụng có giao diện đồ họa trực quan. Bất kể bạn mới bắt đầu hay là một nhà phát triển kinh nghiệm, Tkinter là lựa chọn lý tưởng để xây dựng các ứng dụng GUI đơn giản đến phức tạp.

**Tính năng nổi bật:**

* **Dễ học và sử dụng:** Tkinter cung cấp cú pháp đơn giản và trực quan giúp bạn dễ dàng tạo các giao diện người dùng.
* **Tích hợp sẵn:** Tkinter được tích hợp sẵn trong Python, không yêu cầu cài đặt bổ sung.
* **Đa nền tảng:** Ứng dụng Tkinter có thể chạy trên nhiều hệ điều hành khác nhau.
* **Các widget đa dạng:** Tkinter cung cấp các widget phổ biến như nút, hộp văn bản, danh sách, hộp thoại, vv. để bạn tạo các thành phần giao diện đa dạng.
* **Hỗ trợ tương tác sự kiện:** Tkinter cho phép bạn xử lý các sự kiện như nhấn nút, di chuột và nhiều tương tác khác từ người dùng.
* **Cơ sở hướng dẫn:** Tkinter đi kèm với tài liệu và ví dụ rất phong phú để giúp bạn nắm vững cách sử dụng thư viện.

## 3.3 Một số cơ sở dữ liệu đồ thị

### **3.3.1 Neo4j:**



Neo4j là một hệ quản trị cơ sở dữ liệu đồ thị mã nguồn mở được phát triển bởi Neo4j.Inc. Cơ sở dữ liệu này chủ yếu được thiết kế để lưu trữ và quản lý các dữ liệu có mối quan hệ và liên kết phức tạp. Dữ liệu trong Neo4j được biểu diễn dưới dạng các đỉnh (nodes) và các cạnh (edges) trong đồ thị.

Ngôn ngữ Truy vấn Cypher: Neo4j sử dụng ngôn ngữ truy vấn Cypher, một ngôn ngữ truy vấn đặc trưng dành riêng cho các cơ sở dữ liệu đồ thị. Cypher cho phép bạn tìm kiếm và truy vấn dữ liệu đồ thị bằng cách sử dụng cú pháp dễ đọc và rõ ràng.

Cơ chế Lưu trữ: Neo4j sử dụng cơ chế lưu trữ đặc biệt để lưu trữ dữ liệu đồ thị. Các đỉnh và cạnh được lưu trữ trong các bản ghi dạng b-tree, giúp tối ưu việc truy vấn dữ liệu đồ thị phức tạp.

Mối quan hệ và Thuộc tính: Trong Neo4j, có thể xác định các mối quan hệ giữa các đỉnh và gán thuộc tính cho cả đỉnh và cạnh. Điều này cho phép bạn mô hình hóa các mối quan hệ phức tạp và lưu trữ các thông tin liên quan.

Giao diện người dùng: Neo4j cung cấp giao diện người dùng trực quan giúp thực hiện truy vấn, xem dữ liệu đồ thị và theo dõi cấu trúc dữ liệu một cách dễ dàng..

Ứng dụng sử dụng Neo4j: Neo4j đã được áp dụng trong nhiều lĩnh vực như mạng xã hội, phân tích mạng, quản lý kiến thức, khai phá dữ liệu, hệ thống phân loại, và nhiều ứng dụng khác.

### **3.3.2 Amazon Neptune:**



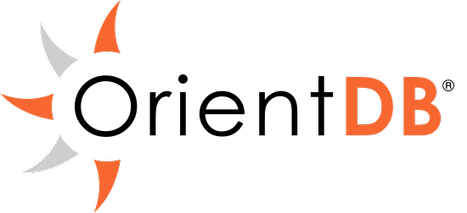
Amazon Neptune là một dịch vụ cơ sở dữ liệu đồ thị quản lý, được xây dựng trên nền tảng của Amazon Web Services (AWS). Dịch vụ này cung cấp một cách tiếp cận dễ dàng để triển khai, quản lý và truy vấn cơ sở dữ liệu đồ thị phức tạp.

### **3.3.3 Microsoft Azure Cosmos DB:**



Microsoft Azure Cosmos DB là một dịch vụ cơ sở dữ liệu đám mây toàn cầu, hỗ trợ nhiều mô hình dữ liệu như cơ sở dữ liệu đồ thị, cơ sở dữ liệu tài liệu, cơ sở dữ liệu từ khoá và cơ sở dữ liệu nhóm cột (column-family database). Dịch vụ này được thiết kế để cung cấp tính linh hoạt, khả năng mở rộng và hiệu suất cao cho các ứng dụng quy mô lớn và phân tán.

**3.3.4 OrientDB: OrientDB**



OrientDB là một hệ quản trị cơ sở dữ liệu đa mô hình mã nguồn mở. Điều này có nghĩa là nó không chỉ hỗ trợ lưu trữ và truy vấn dữ liệu đồ thị, mà còn có thể hoạt động như một cơ sở dữ liệu tài liệu (document database) và cơ sở dữ liệu từ khoá (key-value database) tùy theo cách bạn tổ chức và lưu trữ dữ liệu.

# **Chương 4: Yêu cầu đặt ra và giải quyết yêu cầu**

**4.1 Khảo sát trường**

Sau khi đến khoa tham quan và khảo sát cơ sở hạ tầng, tôi đã nắm bắt được tương đối ổn định về hệ thống mạng của trường. Trường gồm 3 Dãy A B C, 3 tầng lầu bao gồm phòng: Chủ Nhiệm, Văn Phòng Khoa, Đoàn Khoa, Thư Viện, Kỹ Thuật, khoa CNTT, khoa Viễn Thông, khoa Cơ Bản,, khoa QTKD, khoa Điện Tử, Khoa An Toàn Thông Tin, Thí Nghiệm, Thực hành,

Khu vực máy tính văn phòng: 45 máy tính Gồm 5 phòng máy được đặt ip tĩnh, mỗi phòng tương ứng với 1 vlan tương ứng:

* Phòng ban chủ nhiệm: 5 máy tính
* Văn phòng khoa: 15 máy tính
* Đoàn khoa: 10 máy tính
* Phòng kỹ thuật: 10 máy tính
* Quản lý thư viện: 5 máy tính

Phòng máy server: 3 máy tính :

* Server 1 cung cấp dịch vụ dns, mail, web.
* Server 2 lưu trữ database, lưu trữ thong tin sinh viên…
* Server 3 cung cấp dịch vụ domain.

Phòng khoa 60 máy:

* Khoa công nghệ thông tin 10 máy.
* Khoa viễn thông10 máy.
* Cơ bản10 máy.
* Quản trị kinh doanh 10 máy.
* Điện tử10 máy
* An toàn thông tin 10 máy

Khu vực phòng máy tính thực hành và phòng thí nghiệm: 180 máy tính gồm 6 phòng máy thực hành và 3 phòng thí nghiệm

## 4.2 Yêu cầu hệ thống

Hệ thống có thể đăng nhập với 2 quyền:

* Admin: toàn quyền trên hệ thống
* User: chỉ có thể xem thông tin của PC, Switch

User có thể tự đăng kí tài khoản

Admin có thể tạo tài khoản cho admin khác

Mật khẩu phải được mã hóa

Hệ thống có thể thêm xóa sửa các thông tin PC

Hệ thống có thể thêm xóa sửa các thông tin Switch

Thiết lập ràng buộc mỗi switch có 24 cổng – kết nối tối đa 24 PC

Khi thêm hoặc sửa địa chỉ IP phải nằm trong dải network

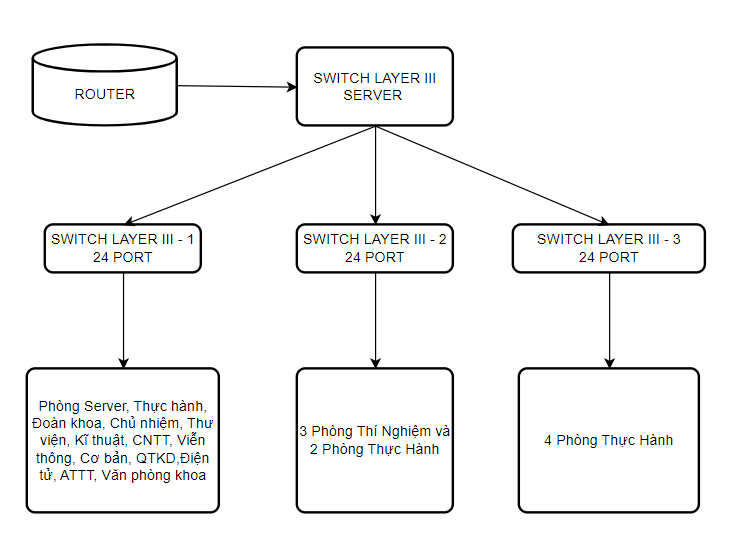
Hệ thống có thông tin về các phòng chứa các PC

**4.3 Yêu cầu thiết kế**

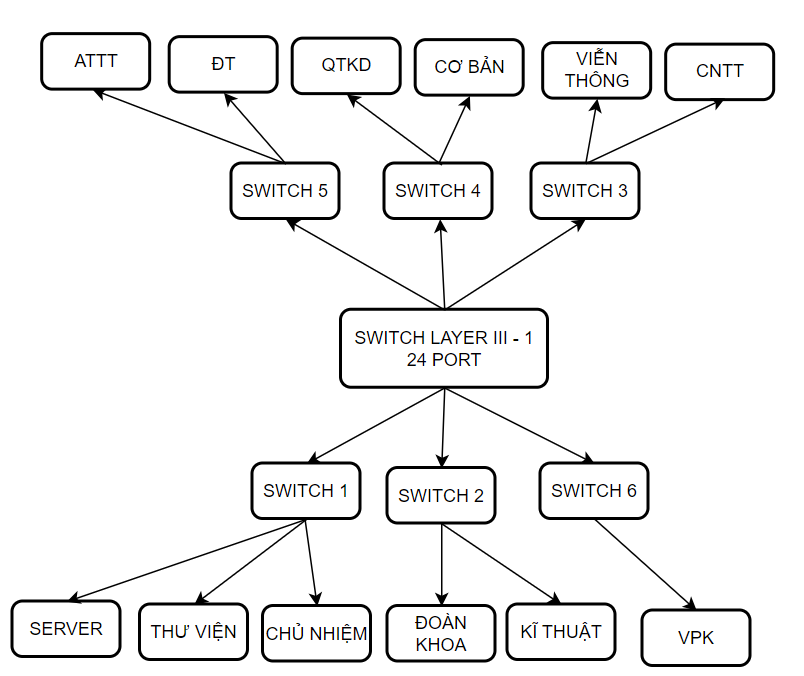
Do địa hình và vị trí thực tế các phòng ban nằm xa nhau nên bố trí mạng hình sao, gồm các switch layer 3 đặt tai trung tâm mạng.

# **Chương 5: Triển khai thiết kế**

## 5.1 Sơ đồ logic

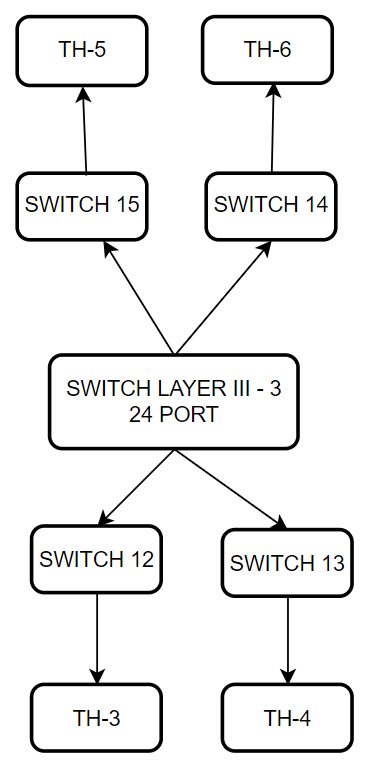
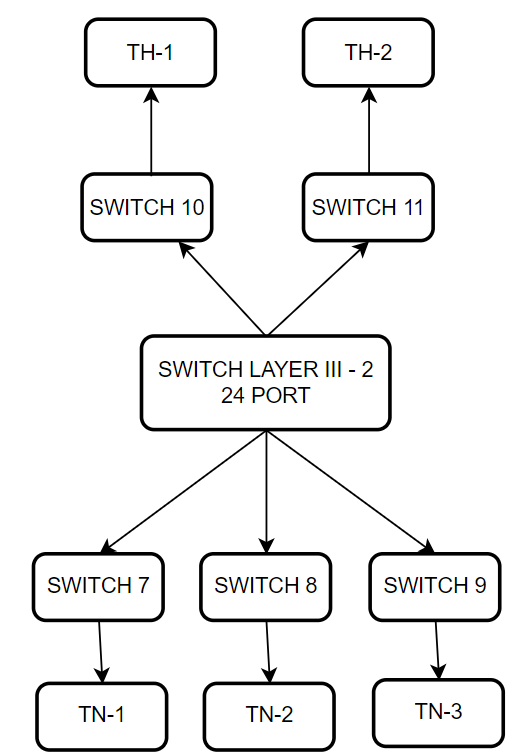


Hình sơ đồ logic thiết kế mạng



Hình Sơ đồ logic của Switch Layer 1

Sơ đồ logic của switch layer 2 Sơ đồ logic của switch layer 3



Hình Sơ đồ logic của Switch Laye 3 Hình Sơ đồ logic của Switch Layer 2

Theo sơ đồ trên ta tiến hành chia subnet theo bảng sau để thiết lập dữ liệu ban đầu:

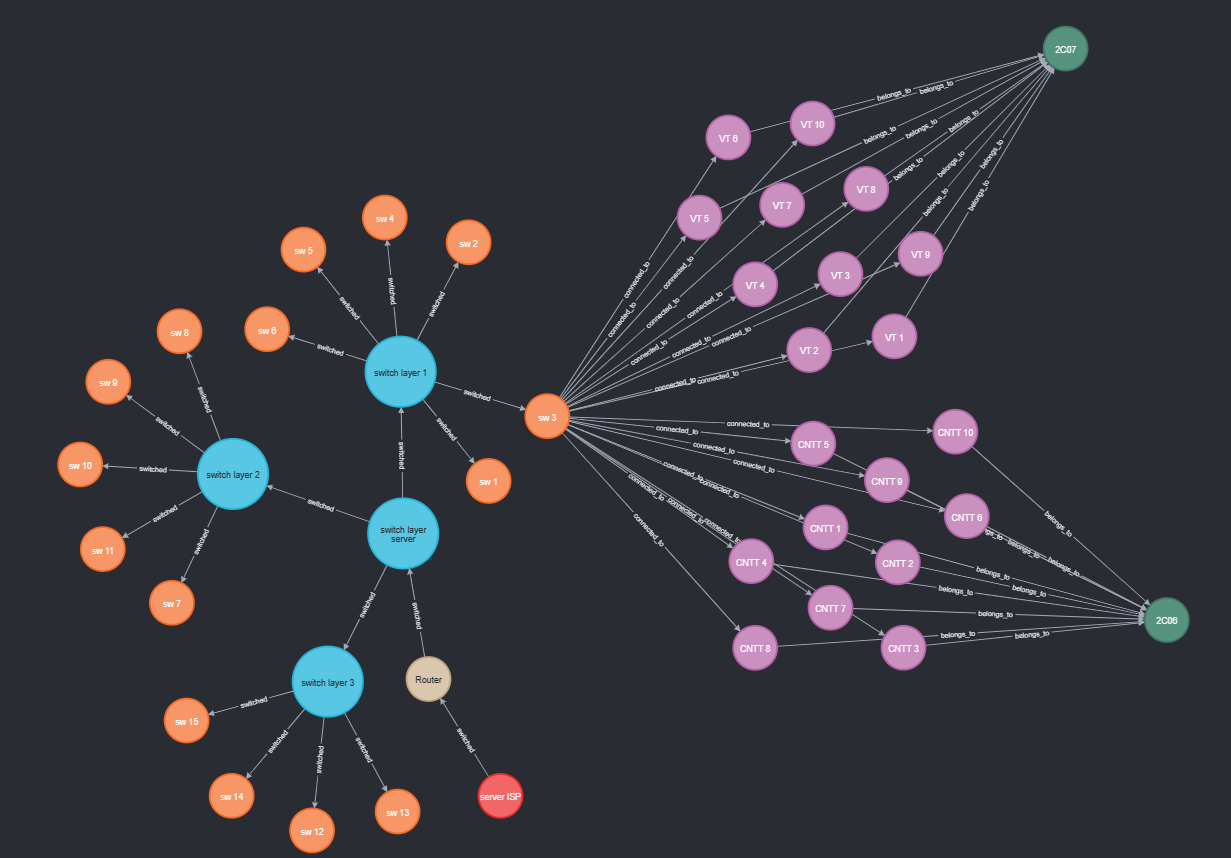
|  |  |  |  |
| --- | --- | --- | --- |
| Host | Ip Range | Netsmask | Network |
| Server | 172.16.0.1 – 172.16.0.6 | 255.255.255.248 | 172.16.0.0/29 |
| Chủ Nhiệm | 172.16.0.9 – 172.16.0.14 | 255.255.255.248 | 172.16.0.8/29 |
| Thư Viện | 172.16.0.15 – 172.16.0.22 | 255.255.255.248 | 172.16.0.16/29 |
| Đoàn Khoa | 172.16.0.33 – 172.16.0.46 | 255.255.255.240 | 172.16.0.32/28 |
| Kĩ Thuật | 172.16.0.49 – 172.16.0.62 | 255.255.255.240 | 172.16.0.48/28 |
| Khoa CNTT | 172.16.0.65 – 172.16.0.78 | 255.255.255.240 | 172.16.0.64/28 |
| Viễn Thông | 172.16.0.81 – 172.16.0.94 | 255.255.255.240 | 172.16.0.80/28 |
| Cơ Bản | 172.16.0.97 – 172.16.0.110 | 255.255.255.240 | 172.16.0.96/28 |
| QTKD | 172.16.0.113 – 172.16.0.126 | 255.255.255.240 | 172.16.0.112/28 |
| Điện Tử | 172.16.0.129 – 172.16.0.142 | 255.255.255.240 | 172.16.0.128/28 |
| ATTT | 172.16.0.145 – 172.16.0.158 | 255.255.255.240 | 172.16.0.144/28 |
| VP Khoa | 172.16.1.1 – 172.16.1.30 | 255.255.255.224 | 172.16.1.0/27 |
| Thí Nghiệm - 1 | 172.16.1.33 – 172.16.1.62 | 255.255.255.224 | 172.16.1.32/27 |
| Thí Nghiệm – 2 | 172.16.1.65 – 172.16.1.94 | 255.255.255.224 | 172.16.1.64/27 |
| Thí Nghiệm – 3 | 172.16.1.97 – 172.16.1.126 | 255.255.255.224 | 172.16.1.96/27 |
| Thực Hành – 1 | 172.16.1.129 – 172.16.1.158 | 255.255.255.224 | 172.16.1.128/27 |
| Thực Hành – 2 | 172.16.1.161 – 172.16.1.190 | 255.255.255.224 | 172.16.1.160/27 |
| Thực Hành – 3 | 172.16.1.193 – 172.16.1.222 | 255.255.255.224 | 172.16.1.192/27 |
| Thực Hành – 4 | 172.16.1.225 – 172.16.1.254 | 255.255.255.224 | 172.16.1.224/27 |
| Thực Hành – 5 | 172.16.2.1 – 172.16.2.30 | 255.255.255.224 | 172.16.2.0/27 |
| Thực Hành – 6 | 172.16.2.33 – 172.16.2.62 | 255.255.255.224 | 172.16.2.32/27 |

Hình Bảng subet được chia theo sơ đồ

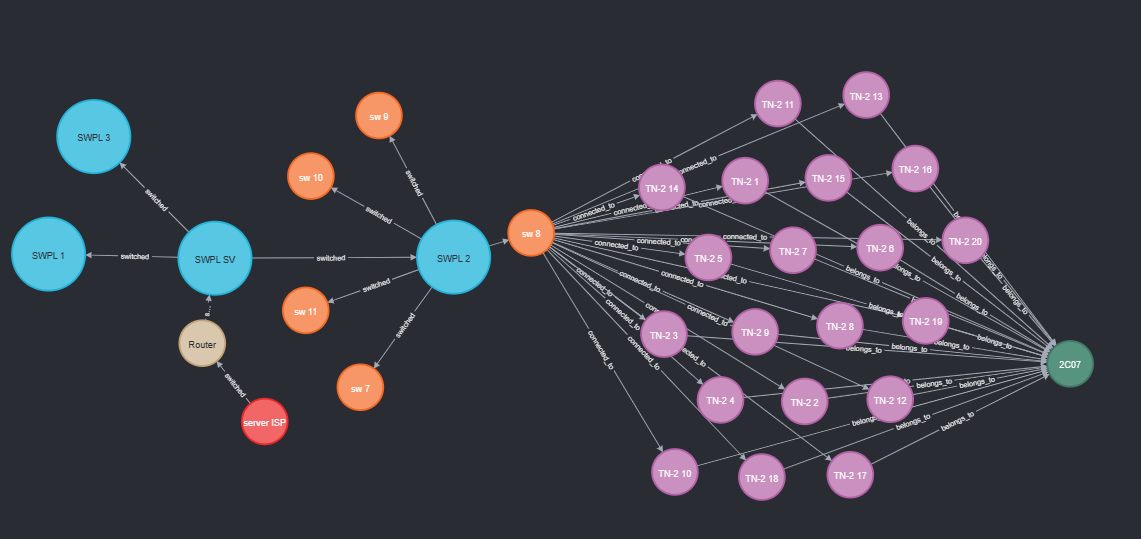
## 5.2 Thiết kế csdl



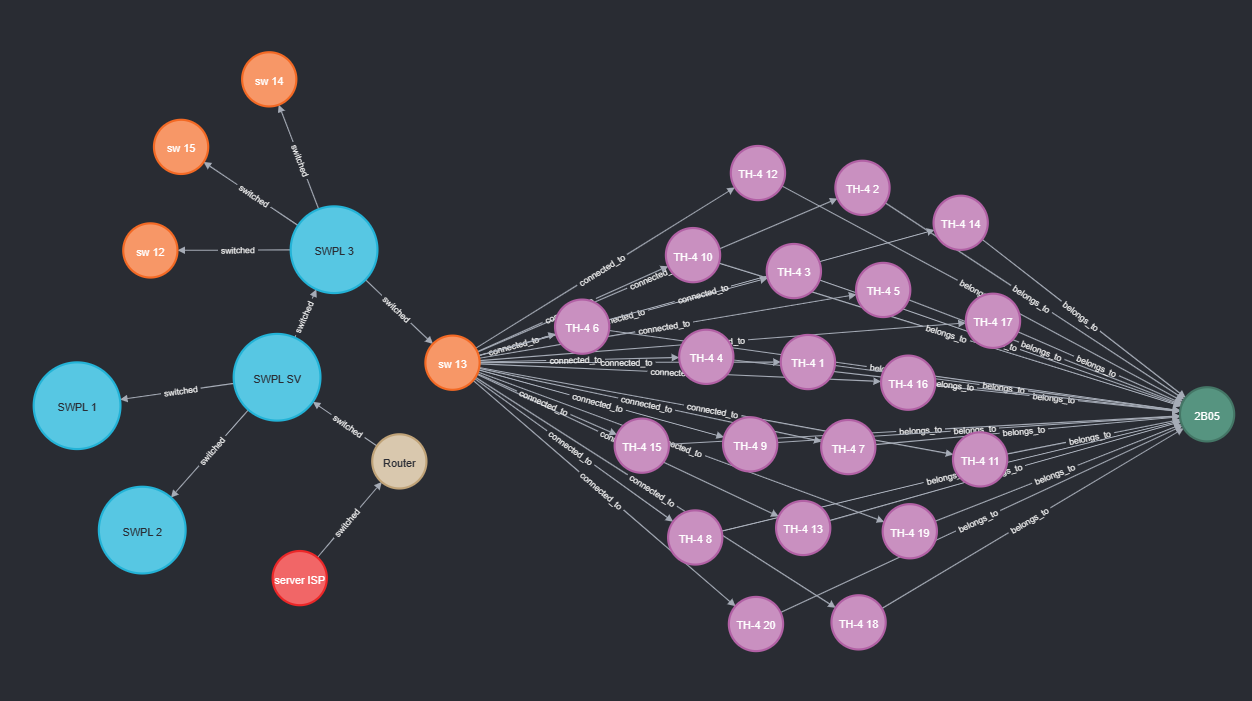
Hình Dữ liệu ở switch 1 của switch layer 1



Hình Dữ liệu ở Switch 3 của Switch layer 1

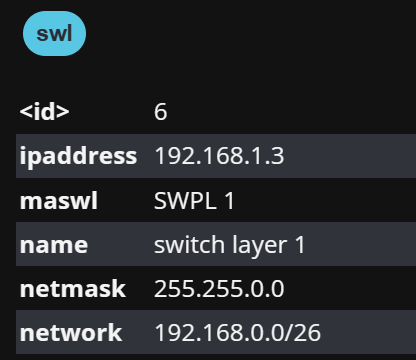


Hình Dữ liệu ở Switch 8 của Switch layer 2



Hình Dữ liệu ở Switch 13 của Switch layer 3

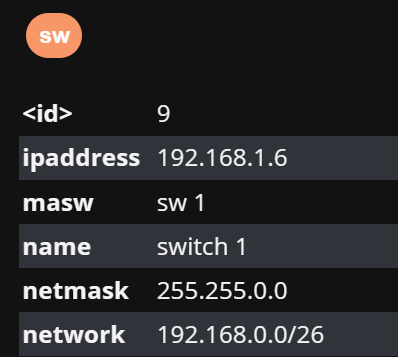
**Mô tả cơ sở dữ liệu**



Hình Bảng dữ liệu node switch layer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | Maswl | String | PK | Mã switch layer |
| 2 | Name | String | Unique,not null | Tên switch layer |
| 3 | Ipaddress | String | Unique, not null | Ip của switch layer |
| 4 | Netmask | String | not null | Netmask của switch layer |
| 5 | Network | String | not null | Network của switch layer |

**Bảng dữ liệu node switch**



Hình Bảng dữ liệu node switch

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | Masw | String | PK | Mã switch |
| 2 | Name | String | Unique, not null | Tên switch |
| 3 | Ipaddress | String | Unique, not null | Ip của switch |
| 4 | Netmask | String | not null | Netmask của switch |
| 5 | Network | String | not null | Network của switch |

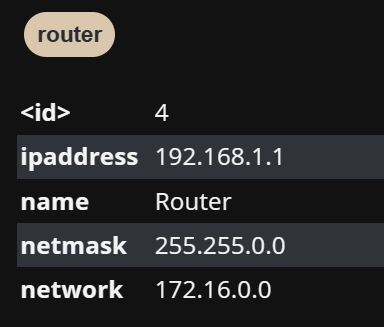
**Bảng dữ liệu node server**



Hình Bảng dữ liệu node server

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | Masv | String | PK | Mã server |
| 2 | Name | String | Unique, not null | Tên server |
| 3 | Ipaddress | String | Unique, not null | Ip của switch layer |
| 4 | Netmask | String | not null | Netmask của server |
| 5 | Network | String | not null | Network của server |

**Bảng dữ liệu node router**



Hình Bảng dữ liệu node router

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | Name | String | Unique, not null | Tên router |
| 2 | Ipaddress | String | Unique, not null | Ip của router |
| 3 | Netmask | String | not null | Netmask của router |
| 4 | Network | String | not null | Network của router |

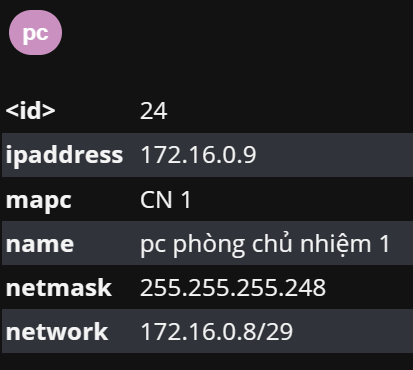
**Bảng dữ liệu node room**



Hình Bảng dữ liệu node room

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | Maphong | String | PK | Mã phòng |
| 2 | Name | String | Unique, not null | Tên phòng |
| 3 | Location | String | Unique, not null | Vị trí của phòng |
| 4 | Roomtype | String | not null | Loại ph |

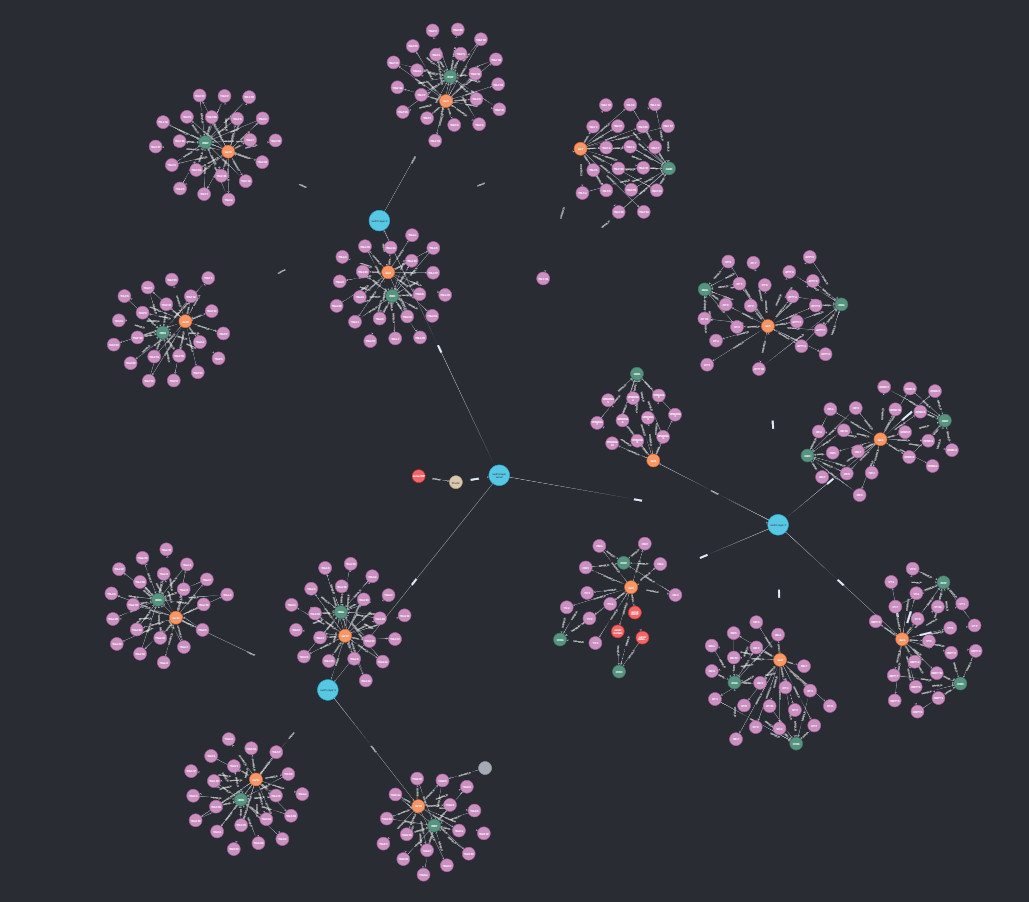
**Bảng dữ liệu node PC**



Hình Bảng dữ liệu node PC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | Mapc | String | PK | Mã pc |
| 2 | Name | String | Unique, not null | Tên pc |
| 3 | Ipaddress | String | Unique, not null | Ip của pc |
| 4 | Netmask | String | not null | Netmask của pc |
| 5 | Network | String | not null | Network của pc |

Cơ sở dữ liệu của chương trình

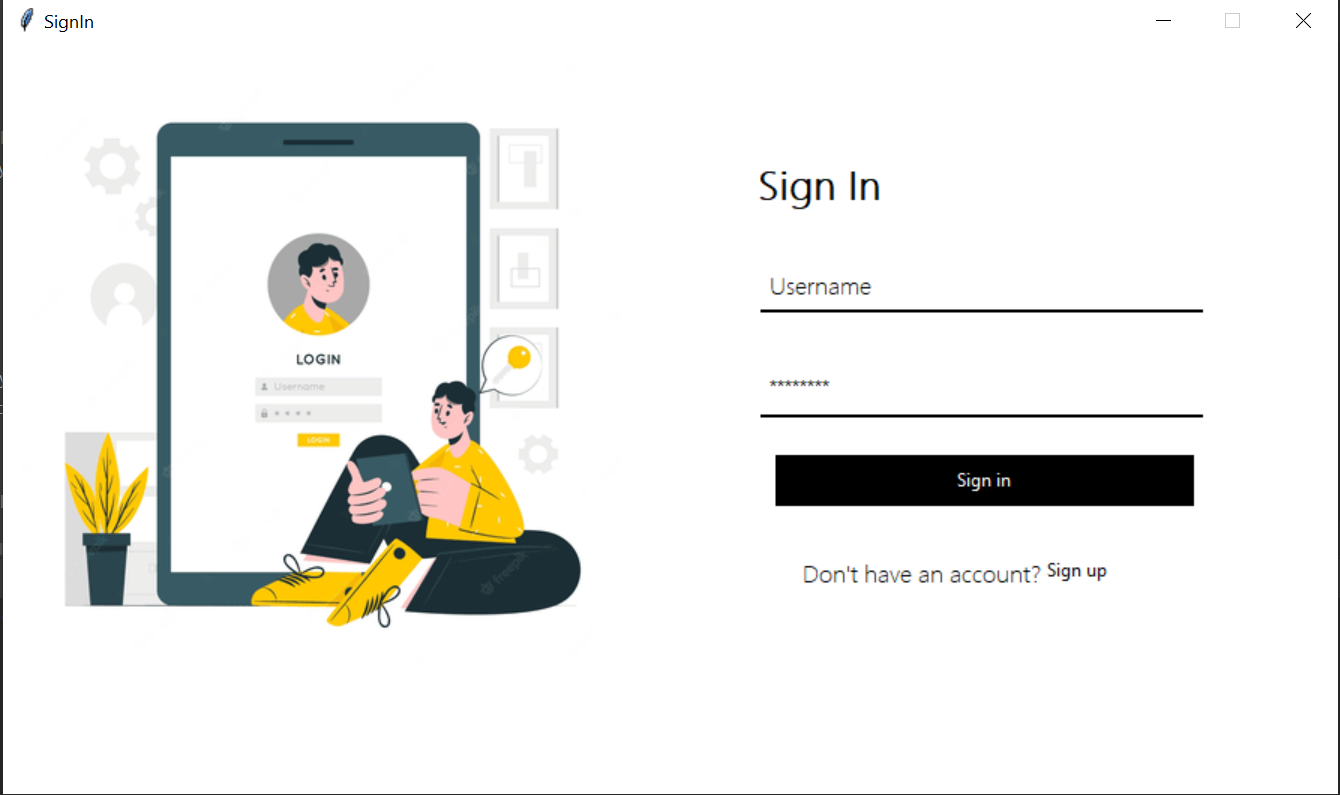


Hình dữ liệ trong hệ thống

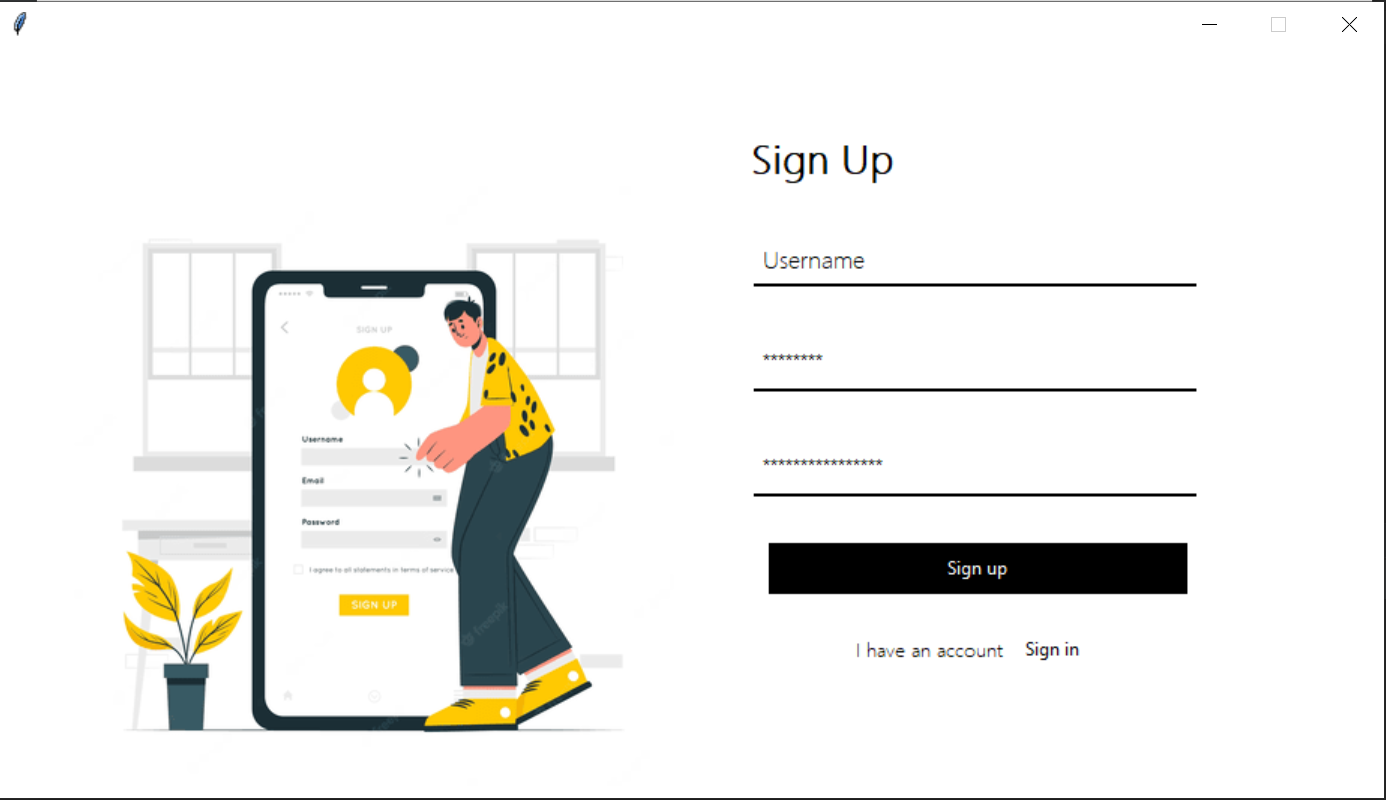
****

Hình các ràng buộc trong csdl

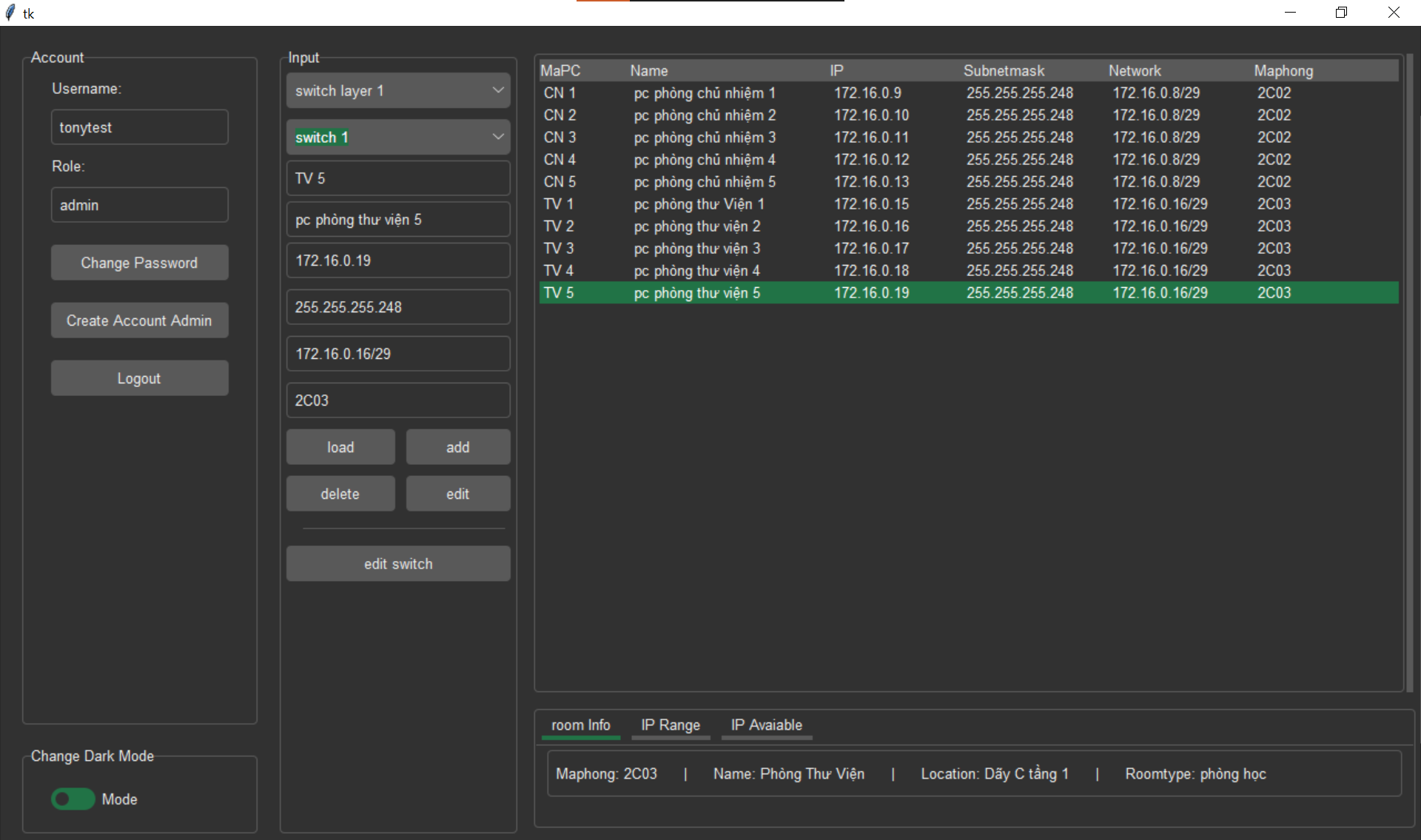
# **Chương 6: Demo ứng dụng đã xây dựng**



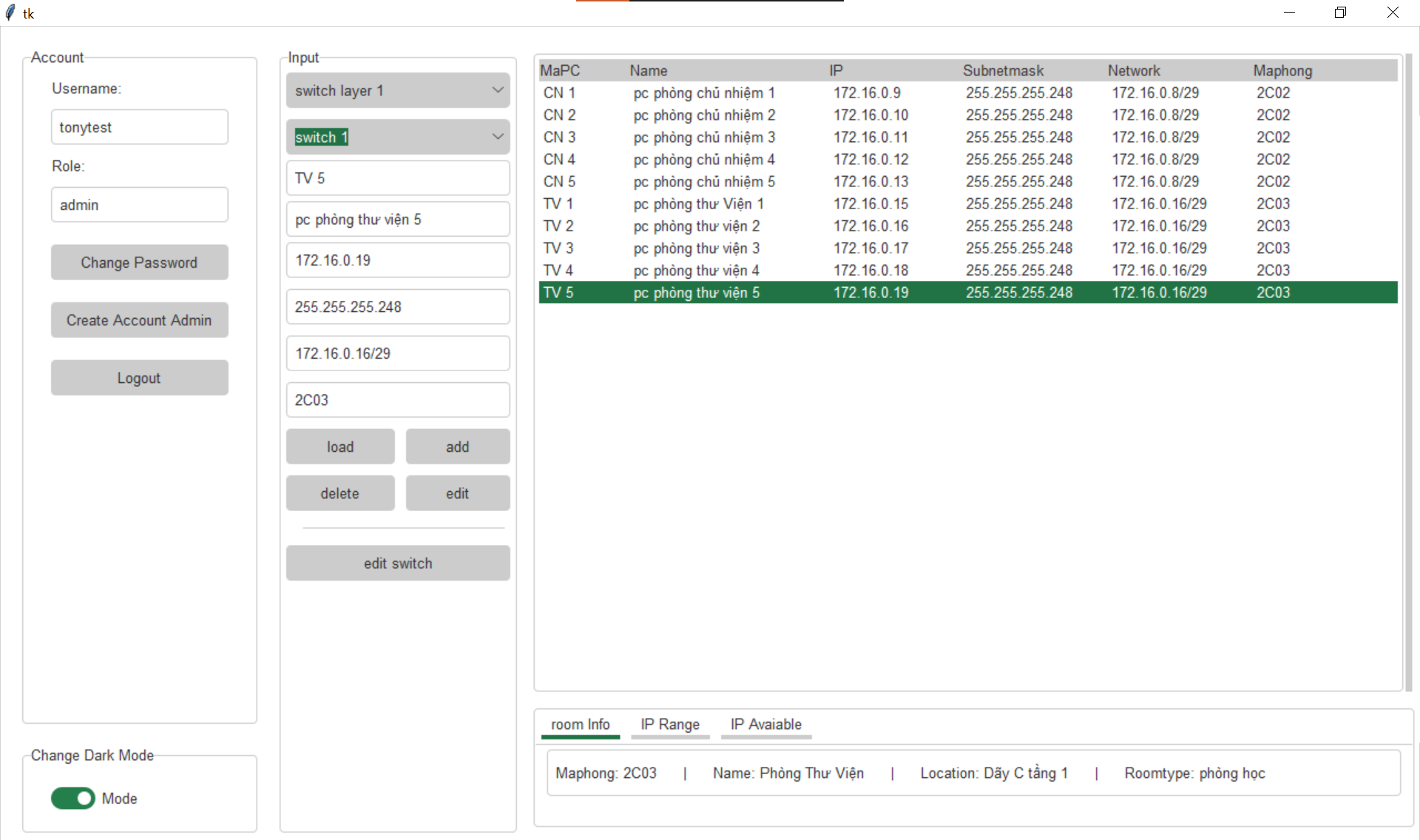
Hình Giao diện đăng nhập

****

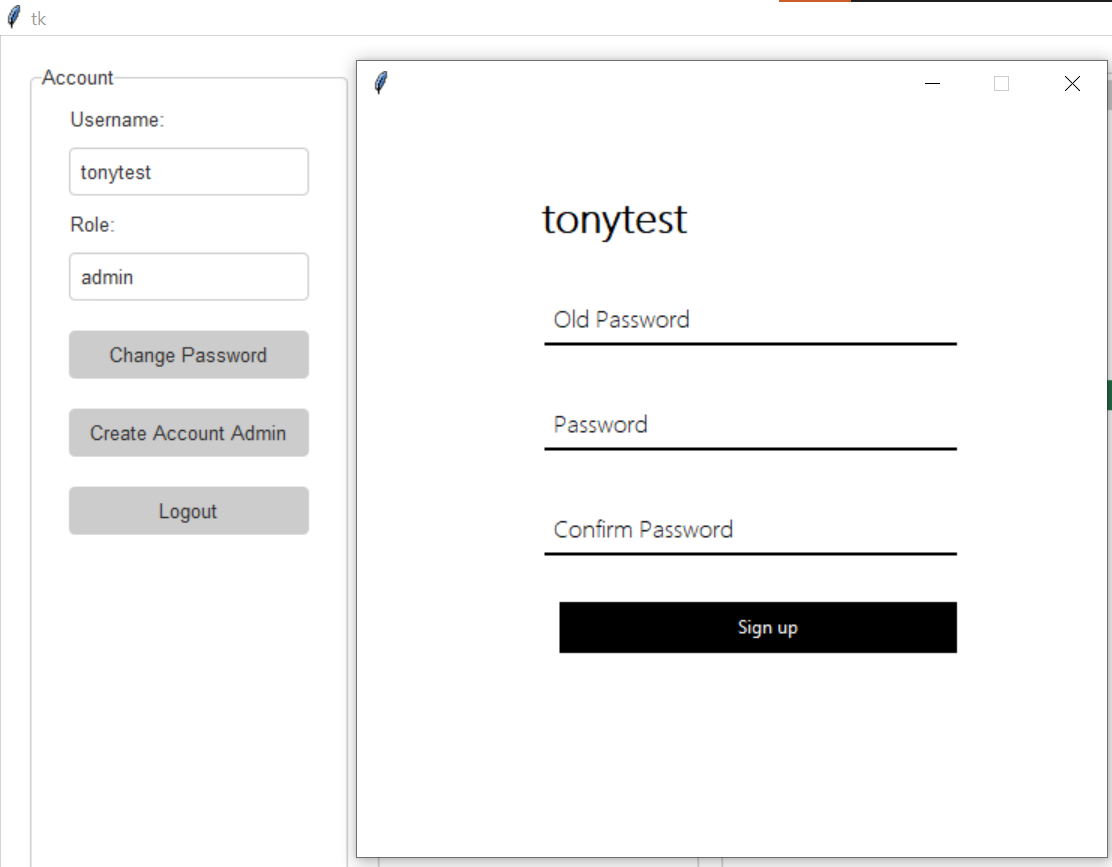
Hình Giao diện đăng ký: (quyền mặc định là User)

****

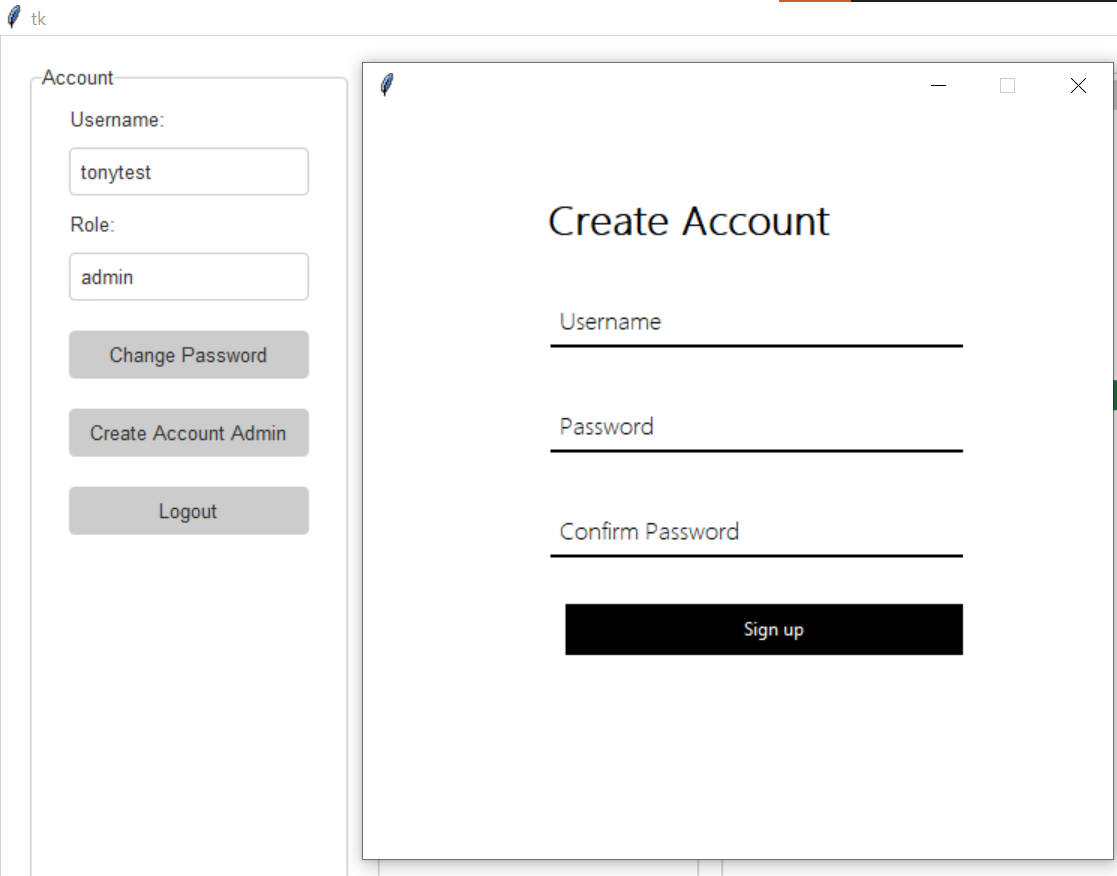
Hình Giao diện Chính Khi load Thành Công PC

****

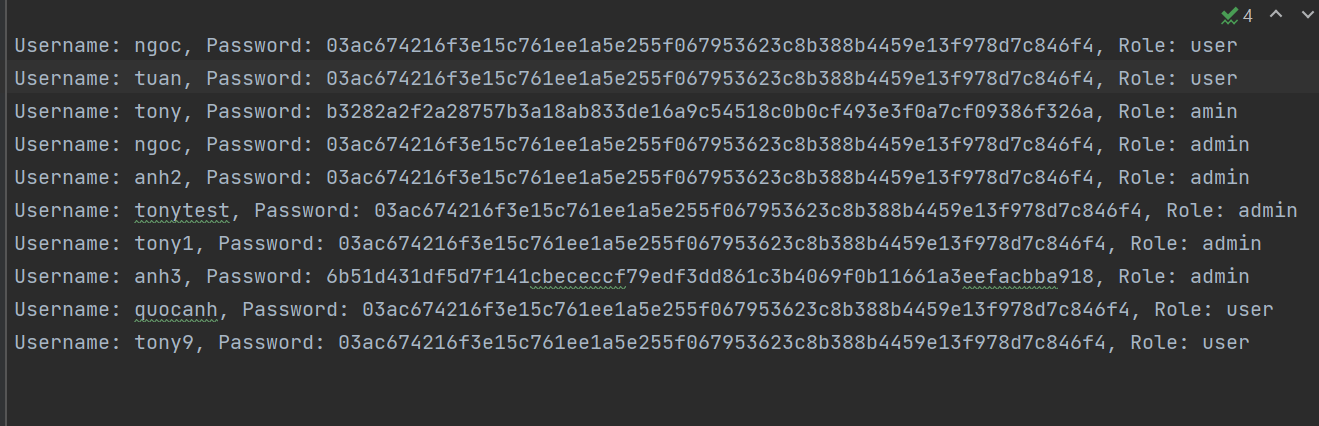
Hình Giao diện khi chuyển chế độ sáng

****

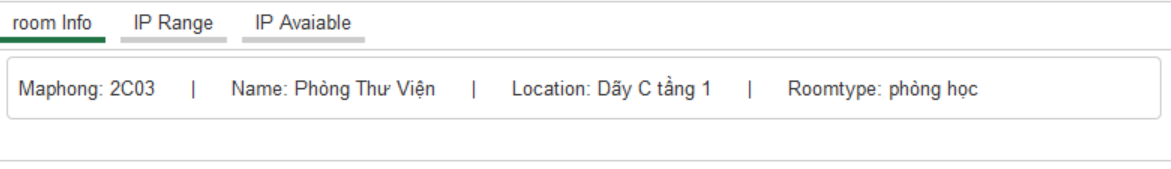
Hình Giao diện đổi mật khẩu

****

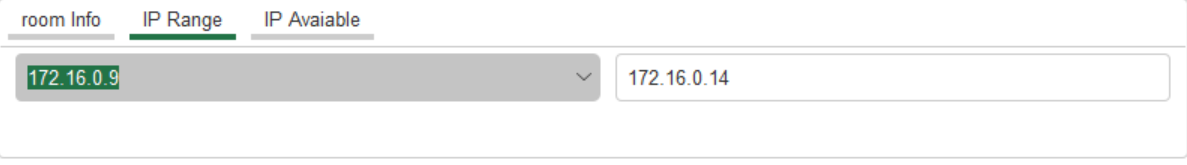
Hình Giao diện Tạo Tài Khoản Admin



Hình File mật khẩu username và quyền:

****

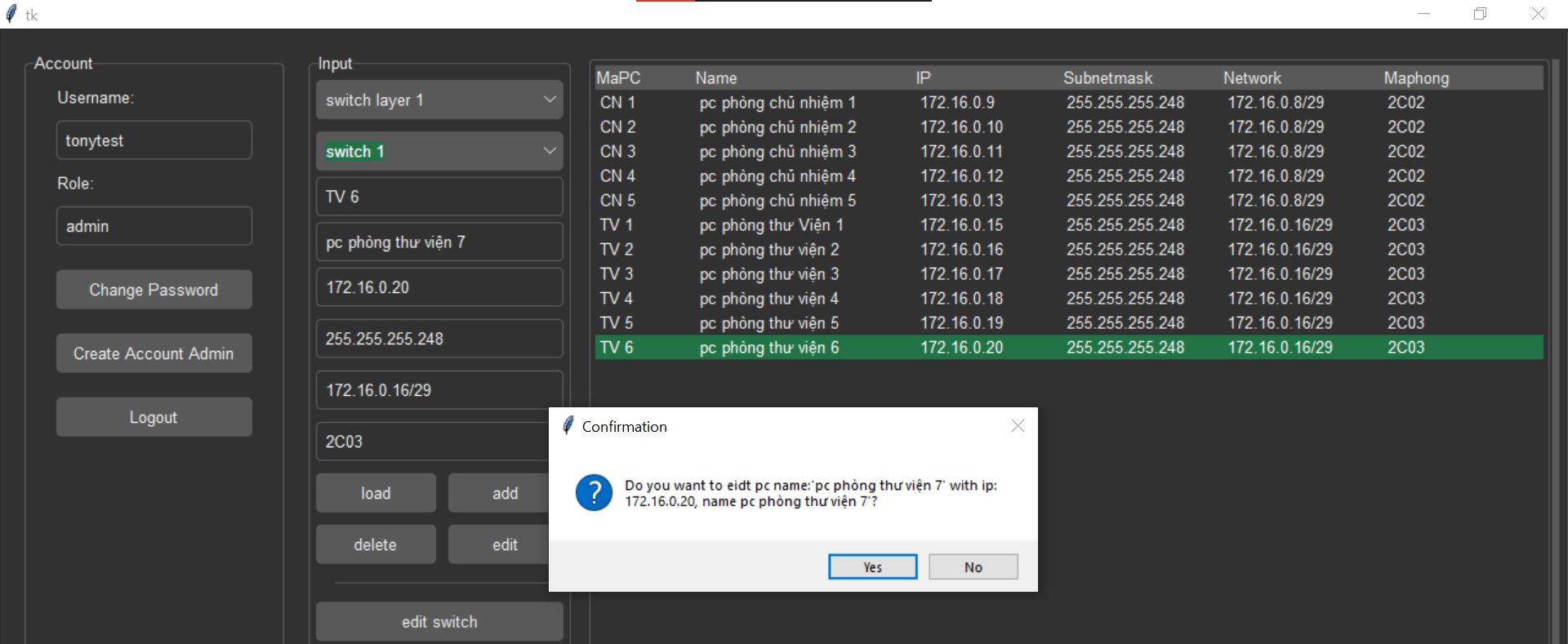
Hình Thanh phụ hiển thị thông tin Phòng



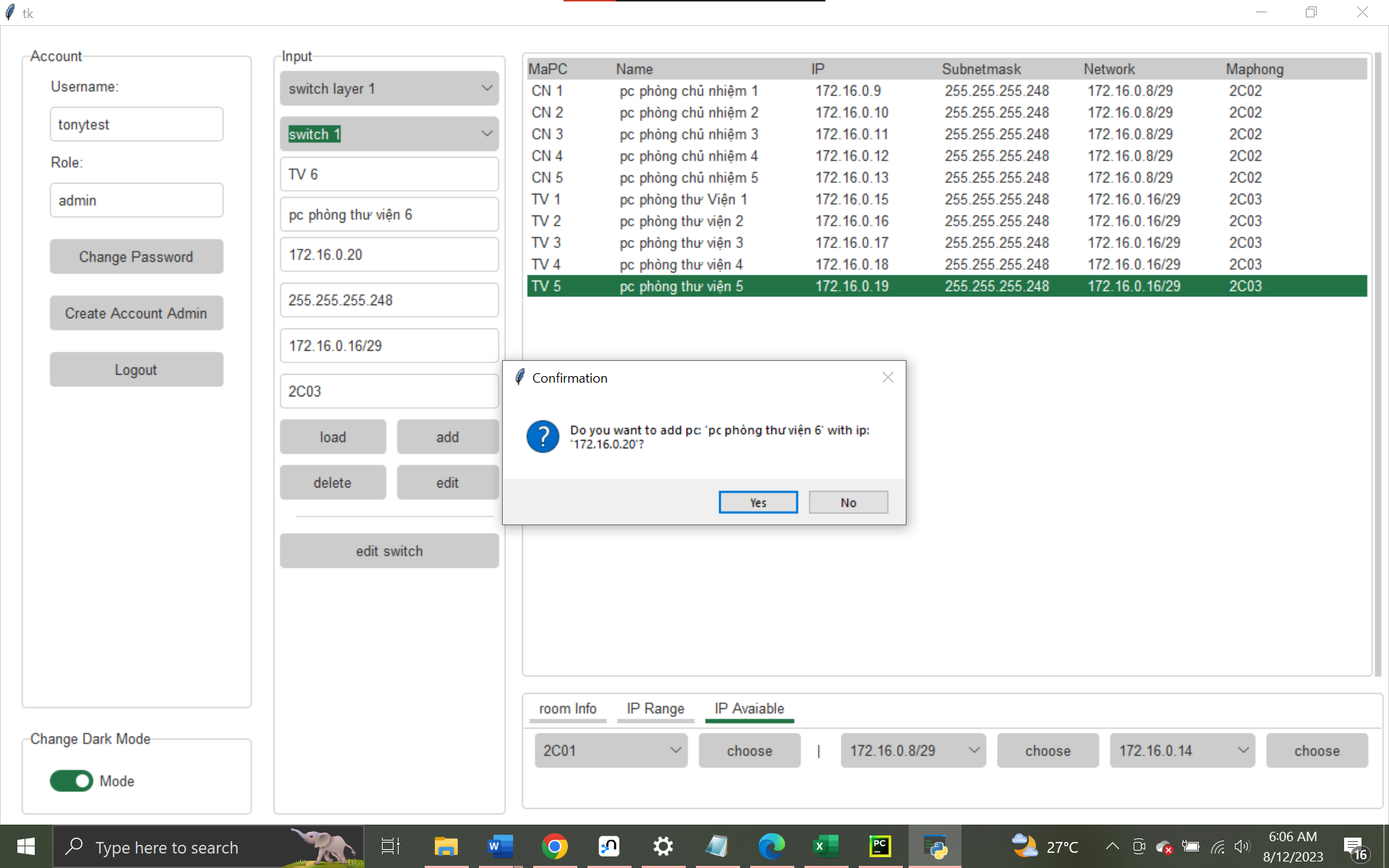
Hình Thanh phụ hiển thị ip range



Hình Thanh Phụ hiển thị Maphong khả dụng, Ip khả dụng và Network khả dụng



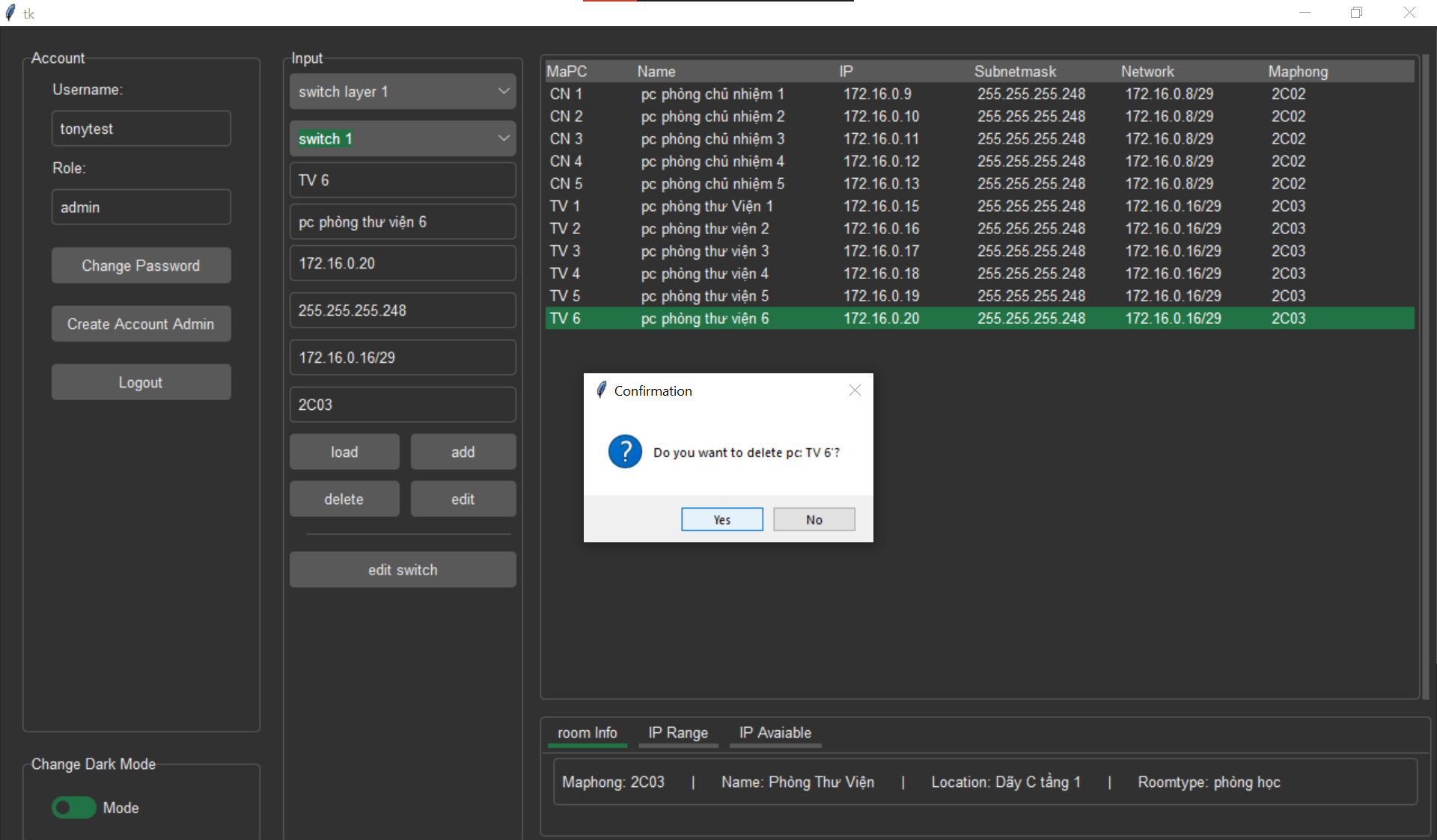
Hình Giao diện khi edit PC



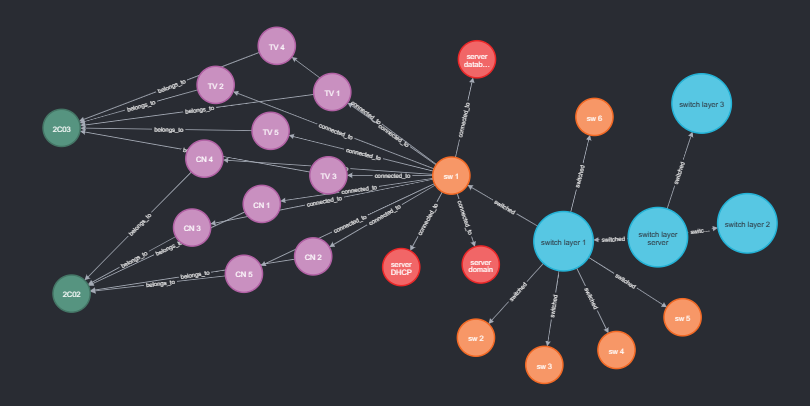
Hình Tính năng thêm PC vào switch

****

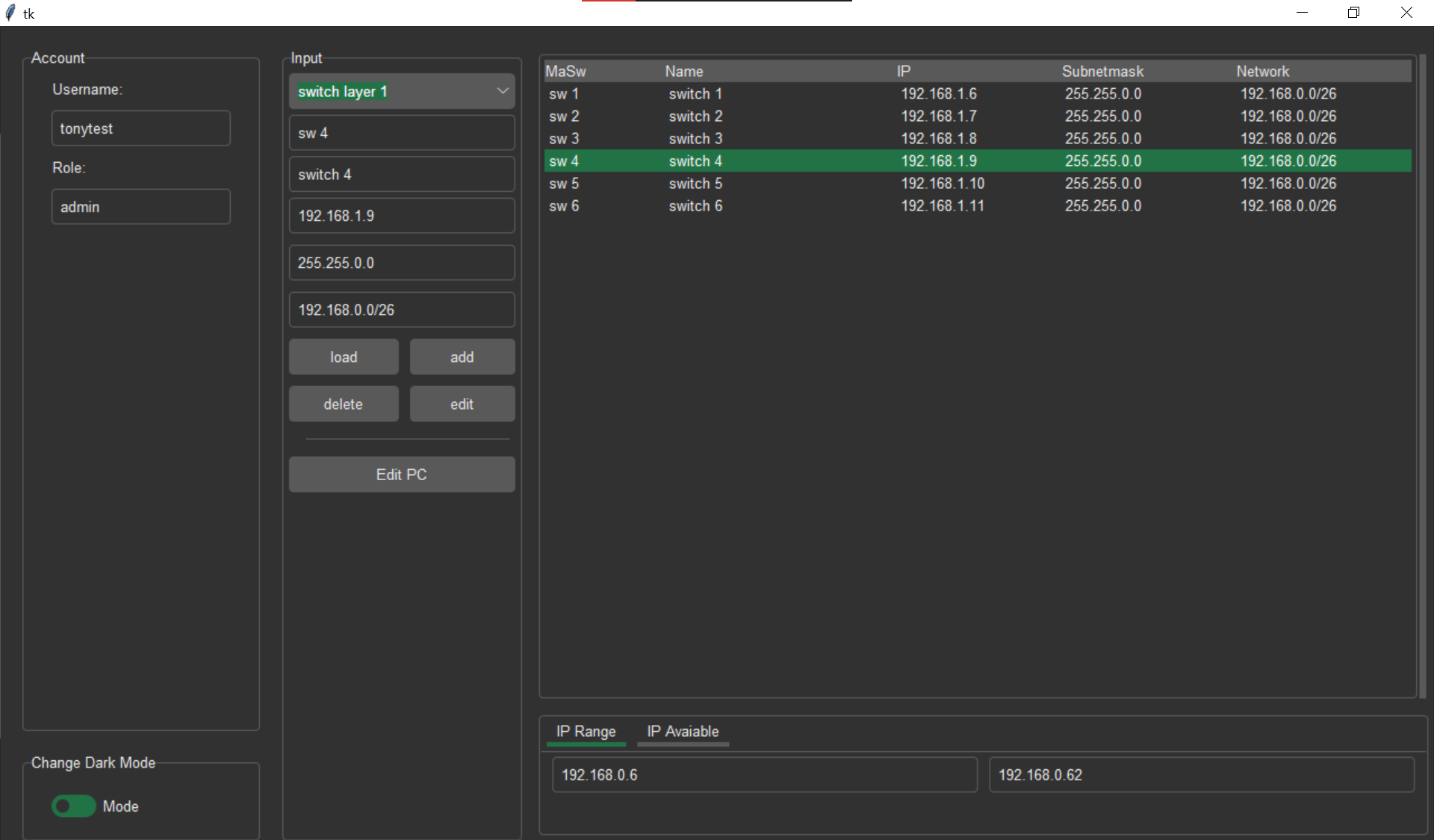
Hình Cơ sở dữ liệu sau khi thêm PC có mã TV 6

****

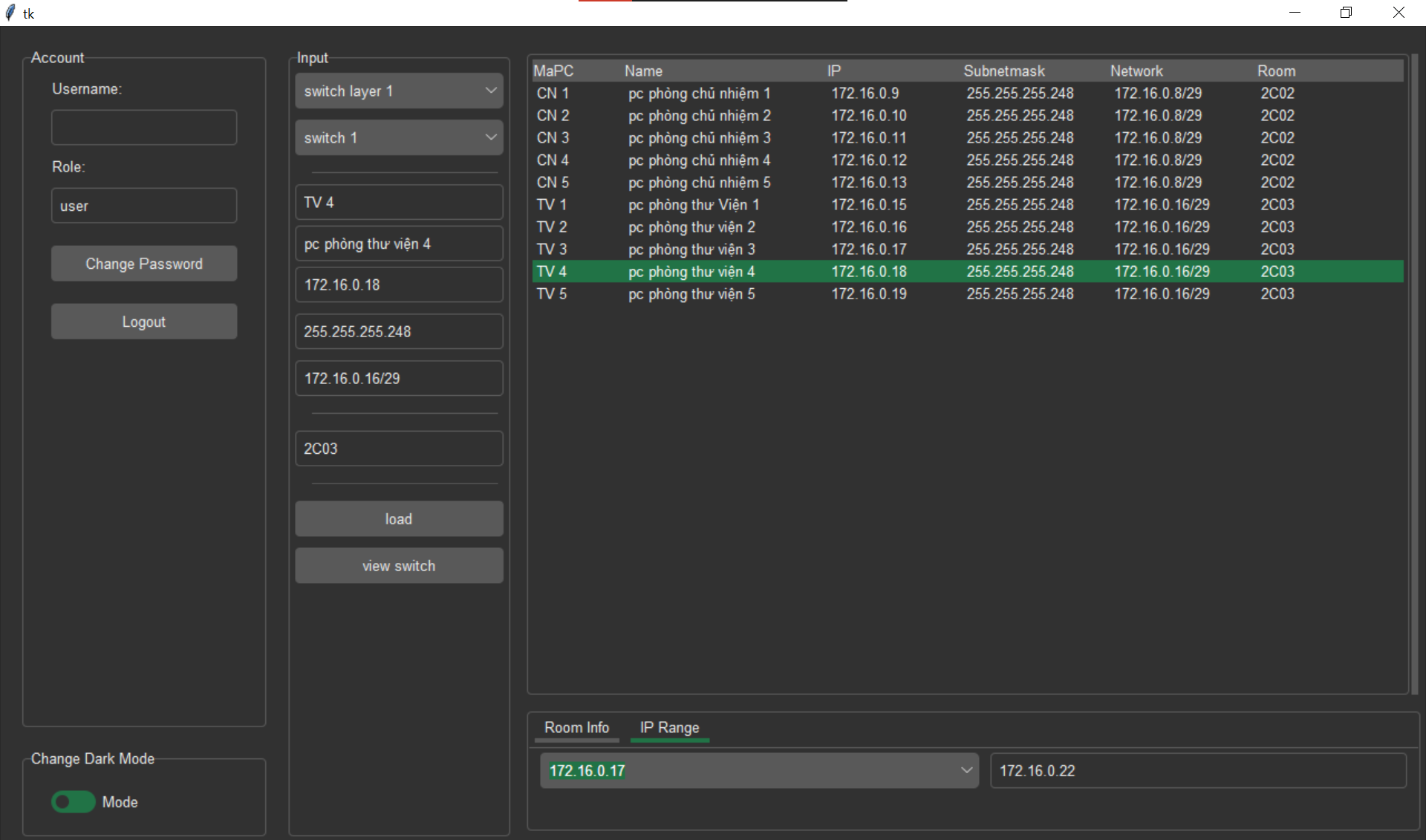
Hình Giao diện khi xóa PC

****

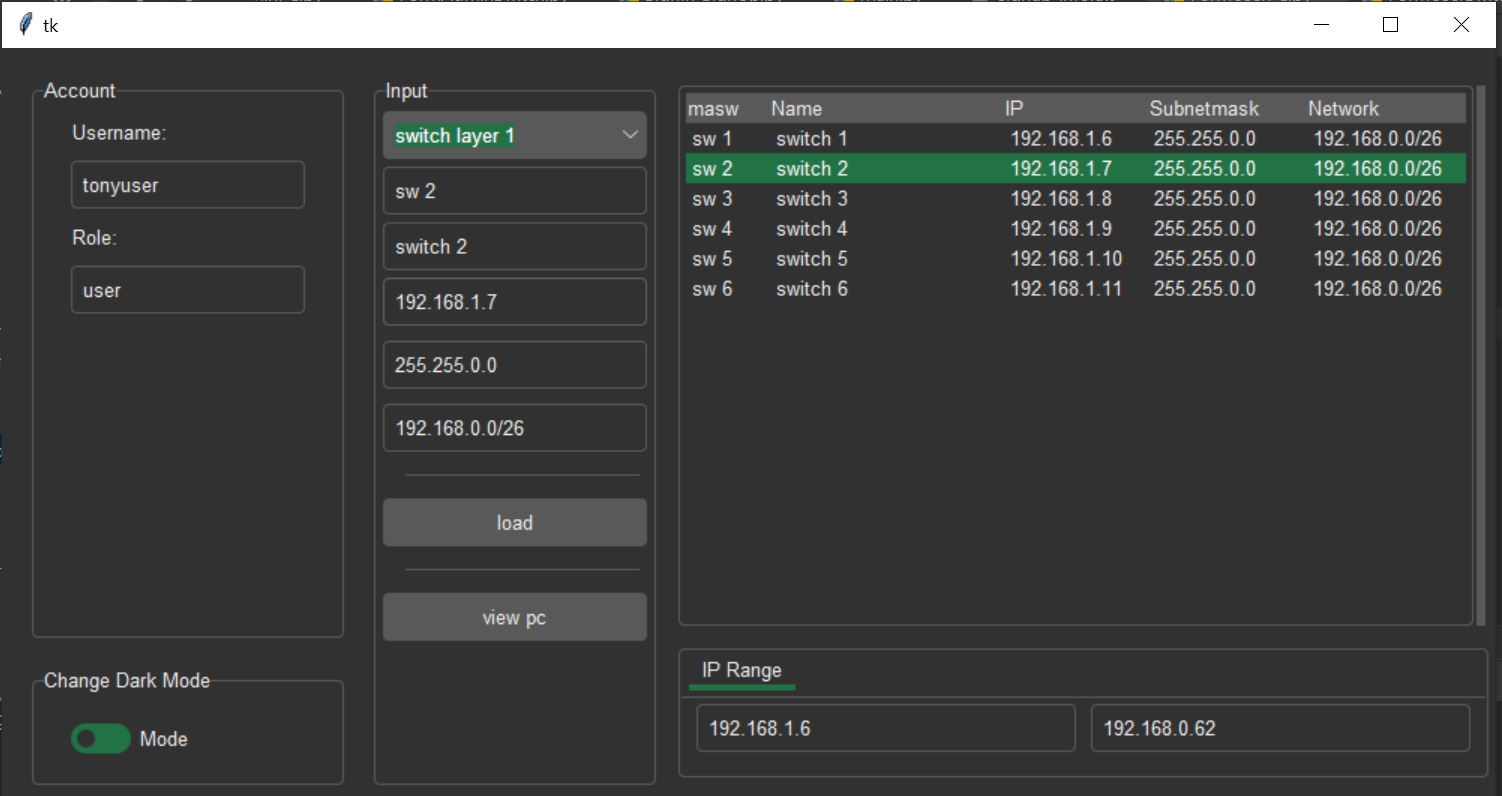
Hình Cơ sở dữ liệu sau khi xóa PC

****

Hình Giao diện admin switch có thể thêm xóa sửa switch

****

Hình Giao diện User PC có thể xem thông tin PC

****

Hình Giao diện User Switch có thể xem thông tin switch

# **Kết luận và đánh giá**

Trong thời gian nghiên cứu và vận dụng những kiến thức đã học ở nhà trường, em đã xây dựng ứng dụng quản lý thông tin mạng nội bộ (Local Network) cho trường đại học, được thiết kế theo TOPO hinh sao. Hệ thống máy chủ đặt tại trung tâm mạng và cung cấp các công cụ cho việc quản trị hệ thống.. Trong bài báo cáo này em đã trình bài một số khái niệm cơ bản về mạng máy tính, cơ sở dữ liệu đồ thị nhằm hoàn thiện hơn về ứng dụng quản lý thông tin mạng thông dụng nhất hiện nay. Em hy vọng những khái niệm này có thể áp dụng cho các lĩnh vực mạng máy tính nói riêng và các ngành liên quan đến công nghệ thông tin nói chung.

Trong quá trình thực hiện niên luận, em đã nhận được sự hướng dẫn tận tình của thầy TS. NGUYỄN HỒNG SƠN giúp em hoàn thành tốt bài niên luận này. Vì kinh nghiệm thực tế còn hạn hẹp nên niên luận không tránh khỏi những sai sót, rất mong nhận được sự góp ý nhiều hơn của quý thầy cô.

# **DANH MỤC TÀI LIỆU THAM KHẢO**

1. Ian Robinson, Jim Webber, and Emil Eifrem, 2013, “**Graph Databases**”, O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.
2. <https://voer.edu.vn/m/mang-may-tinh-computer-networks/a77d0a8e>
3. <https://realpython.com/python-gui-tkinter>
4. <https://neo4j.com/developer/cypher/querying/>
5. The Neo4j Team neo4j.org, 2013, **“The Neo4j Manual”**
6. https://voer.edu.vn/c/lien-mang/b14d14a4/13d0eb88