|  |
| --- |
| TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN **KHOA CÔNG NGHỆ THÔNG TIN**  **BỘ MÔN CÔNG NGHỆ PHẦN MỀM**  **ĐẶNG ANH HÀO – 1612173**  **PHẠM ANH QUỐC – 1612543**  TÌM HIỂU VÀ XÂY DỰNG HỆ THỐNG  HỖ TRỢ THÔNG TIN BẤT ĐỘNG SẢN  **KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN CNTT**  **GIÁO VIÊN HƯỚNG DẪN** **ThS. TRẦN VĂN QUÝ**  **KHÓA 2016 - 2020** |

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

Khóa luận đáp ứng yêu cầu của Khóa luận cử nhân CNTT.

TpHCM, ngày ….. tháng …… năm ……

Giáo viên hướng dẫn

[Ký tên và ghi rõ họ tên]

**NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN**

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

Khóa luận đáp ứng yêu cầu của Khóa luận cử nhân CNTT.

TpHCM, ngày ….. tháng …… năm ……

Giáo viên phản biện

[Ký tên và ghi rõ họ tên]

Lời cảm ơn

Khoa Công Nghệ Thông Tin

Bộ môn …………………….

**ĐỀ CƯƠNG CHI TIẾT**

|  |
| --- |
| **Tên Đề Tài:** Tìm hiểu và xây dựng hệ thống hỗ trợ thông tin bất động sản |
| **Giáo viên hướng dẫn:** Th.S Trần Văn Quý |
| **Thời gian thực hiện:** Từ ngày 01/03/2020 đến 15/07/2020 |
| **Sinh viên thực hiện:** Đặng Anh Hào – 1612173  Phạm Anh Quốc – 1612543 |
| **Loại đề tài:** Xây dựng ứng dụng |

|  |  |
| --- | --- |
| **Nội Dung Đề Tài:** (mô tả chi tiết nội dung đề tài, yêu cầu, phương pháp thực hiện, kết quả đạt được,…) | |
| **Kế Hoạch Thực Hiện:** (mô tả chi tiết thời gian của các giai đoạn thực hiện và phân công công việc của từng thành viên trong nhóm) | |
| **Xác nhận của GVHD** | **Ngày…… tháng…… năm……**  **SV Thực hiện**  *Đặng Anh Hào – Phạm Anh Quốc* |

✪✪✪

***MỤC LỤC***

***Danh mục hình vẽ***

***Danh mục bảng biểu***

***Danh mục chữ viết tắt***

***Tóm tắt khoá luận***

Trình bày tóm tắt vấn đề nghiên cứu, các hướng tiếp cận, cách giải quyết vấn đề và một số kết quả đạt được

# MỞ ĐẦU

# NỀN TẢNG CÔNG NGHỆ

## Rest API

### Rest API là gì?

API là một giao diện lập trình ứng dụng (Application programming interface). Đây là một bộ quy tắc cho phép các chương trình giao tiếp được với nhau. Nhà phát triển ứng dụng thiết kế API từ máy chủ (server) và cho phép người dùng (client) kết nối với máy chủ. API có thể trả về dữ liệu mà bạn cần thông qua các kiểu dữ liệu phổ biến như JSON hay XML.

REST được viết tắt của cụm từ “Representational State Transfer”. REST là một bộ quy tắc mà các nhà phát triển API cần tuân thủ theo khi bắt đầu xây dựng API. Một trong những quy tắc là người dùng sẽ có thể có được một phần dữ liệu (resource) khi truy cập một URL (Uniform Resource Locator).

Mỗi URL được gọi là một yêu cầu (request), còn dữ liệu được trả về từ server được gọi là một phản hồi (response). Chức năng quan trọng nhất của REST là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE..) và cách định dạng các URL cho ứng dụng web để quản lý các resoure.

### Nguyên tắc

Dưới đây là 6 nguyên tắc cơ bản trong mô hình REST:

* **Client-server:** Tách biệt mối quan hệ giữa giao diện và dữ liệu, cải thiện tính di động của giao diện người dùng trên nhiều nền tảng và cải thiện khả năng mở rộng bằng cách đơn giản hóa thành phần máy chủ.
* **Phi trạng thái (Stateless):** Mỗi request được gửi từ phía client đến server phải chứa tất cả những thông tin cần thiết để server có thể hiểu được, không lợi dụng các thông tin được lưu trên server. Do đó, session được giữ hoàn toàn trên máy client.
* **Hỗ trợ bộ nhớ đệm (Cacheable):** Các ràng buộc về bộ nhớ đệm yêu cầu rằng dữ liệu trong phản hồi yêu cầu phải được gán nhãn ngầm hoặc rõ ràng là có thể lưu trong bộ nhớ cache hay không. Nếu phản hồi là có thể lưu vào bộ nhớ cache thì bộ nhớ cache của client được sử dụng lại dữ liệu phản hồi đó cho các lần request tương tự trong tương lai.
* **Giao diện thống nhất (Uniform interface):** Uniform interface cho phép client giao tiếp với server bằng một ngôn ngữ duy nhất, độc lập với kiến trúc. Interface này sẽ cung cấp một phương tiện giao tiếp được chuẩn hóa, không thay đổi giữa client và server, chẳng hạn sử dụng HTTP với URI resources, CRUD (Create, Read, Update, Delete) và JSON.
* **Hệ thống phân lớp (Layered System):** Hệ thống phân lớp cho phép một kiến trúc được tạo thành từ các lớp phân cấp, giúp các thành phần độc lập với nhau. Do đó, hệ thống dễ dàng mở rộng.
* **Code on demand:** REST cho phép một số tính năng ở máy client được mở rộng bằng cách thực thi mã dưới dạng applets hoặc script.

## JSON

### Giới thiệu

JSON viết tắt của từ JavaScript Object Notation, đây là một định dạng trao đổi dữ liệu nhẹ. Con người dễ dàng đọc hiểu, viết. Máy tính dễ dàng phát sinh hoặc phân tích. JSON được dự trên một tập hợp con của *JavaScript Programming Language Standard ECM-262 3rd Edition – December 1999.*

JSON là một định dạng văn bản hoàn toàn độc lập với ngôn ngữ nhưng sử dụng các quy ước quen thuộc với các lập trình viên thuộc C-family of languages, bao gồm C, C++, C#, Java, JavaScript, Perl, Python, và nhiều ngôn ngữ khác. Do đó, JSON trở thành một ngôn ngữ lý tưởng dể trao đổi dữ liệu.

### Cách xây dựng

Json được xây dựng dựa trên hai cấu trúc:

* Là một tập hợp các cặp name – value tương tự như các đối tượng trong các ngôn ngữ lập trình.
* Một dãy các giá trị được sắp xếp tương tự như các array, vector, list hay sequence.

### Ứng dụng trong hệ thống

Trong quy mô luận văn này, hai hình thức biểu diễn của JSON được sử dụng là:

* Object: Là một cặp name – value. Name và value được cách nhau bởi dấu hai chấm “:”. Value là một kiểu dữ liệu bất kỳ quy ước.
* Array: Là một dãy các object. Array được bắt đầu bởi dấu ngoặc vuông mở “[“ và kết thúc bằng dấu ngoặc vuông đóng “]”. Các object được cách nhau bởi dấu phẩy “,”.

## Python

### Giới thiệu

Python là một ngôn ngữ lập trình thông dịch, hướng đối tượng, và là một ngôn ngữ bậc cao. Python hỗ trợ các module và package, khuyến khích chương trình module hóa và tái sử dụng mã. Trình thông dịch Python và thư viện chuẩn mở rộng có sẵn dưới dạng mã nguồn hoặc dạng nhị phân miễn phí cho tất cả các nền tảng chính và có thể phân phối tự do. Ngoài ra, python là một ngôn ngữ có khả năng chạy trên nhiều nền tảng như hệ điều hành Windows, Linux/Unix, Mac, OS/2, Amiga và những hiện điều hành khác.

### Đặc điểm

* Rất dễ học và sử dụng.
* Vừa hướng thủ tục (procedural-oriented), vừa hướng đối tượng (object-oriented).
* Hỗ trợ module và hỗ trợ gói (package).
* Xử lý lỗi bằng ngoại lệ (Exception).
* Kiểu dữ liệu động ở mức cao.
* Có các bộ thư viện chuẩn và các module ngoài, đáp ứng tất cả các nhu cầu lập trình.
* Có khả năng tương tác với các module khác viết trên C/C++ (hoặc Java cho Jython, hoặc .Net cho IronPython).
* Có thể nhúng vào ứng dụng như một giao tiếp kịch bản (scripting interface).

### Lịch sử

Python được hình thành vào cuối những năm 1980. Sau đó, nó được triển khai vào tháng 12 năm 1989 bởi Guido Van Rossum tại CWI ở Netherland. Vào tháng 2 năm 1991, phiên bản python 0.9.0 được phát hành. Tính đến ngày 14/10/2019, python đã có phiên bản 3.8.

## Django framework

Django là một Python web framework miễn phí mã nguồn mở, bậc cao, được tạo ra với mục đích giúp quá trình phát triển ứng dụng trở nên nhanh chóng, dễ dàng hơn.

Những tính năng nổi bật của Django:

* Phát triển ứng dụng nhanh chóng: Django được thiết kế giúp cho các nhà phát triển ứng dụng xây dựng ứng dụng một cách nhanh chóng.
* Hỗ trợ mạnh mẽ: Django bao gồm rất nhiều các tính năng, có thể sử dụng để xử lý các các tác vụ phổ biến trên web. Cung cấp các chức năng xác thực, quản trị nội dung, site maps, RSS feeds và nhiều chức năng khác.
* Bảo mật cao: Vấn đề bảo mật được coi trọng, do đó Django giúp các nhà phát triển tránh các vấn đề như SQL injection, cross-site scripting, cross-site request forgery và clickjacking.
* Khả năng mở rộng cao: Được thiết kế với khả năng mở rộng linh hoạt.
* Cực kỳ linh hoạt: Các công ty, tổ chức sử dụng Django để xây dựng các ứng dụng từ hệ thống quản lý nội dung đến mạng xã hội, nền tảng điện toán khoa học.

## MySQL

MySQL là một hệ quản trị cơ sở dữ liệu mã nguồn mở (Relational Database Management System). MySQL được phát triển vào năm 1994 tại Công ty Thụy Điển MySQL AB. Vào 2008, công ty công nghệ Mỹ Sun Microsystem giữ quyền sở hữu MySQL. Đến năm 2010, gã khổng lồ Oracle mua Sun Microsystems và MySQL.

Chúng ta có thể tải về MySQL miễn phí từ trang chủ. MySQL có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS,...

MySQL là một trong những phần mềm có độ phổ biến nhất, được các công ty công nghệ lớn sử dụng như Facebook, Twitter, Google.

Đây là những lý do mà MySQL lại phổ biến rộng rãi:

* Linh hoạt và dễ dùng
* Hiệu năng cao
* Tiêu chuẩn trong ngành
* An toàn

## Cloudinary

Cloudinary là một dịch vụ đám mây, cung cấp giải pháp cho toàn bộ hệ thống quản lý hình ảnh của ứng dụng web. Dễ dàng tải hình ảnh lên cloud, có các chức năng hữu ích như, điều chỉnh kích thước, cắt xén, chuyển đổi hình ảnh.

Bên cạnh đó, Cloudinary còn cung cấp API dựa trên URL và HTTP có thể dễ dàng tích hợp với nhiều ngôn ngữ lập trình như: Ruby, PHP, Python, Node.js, Java, .Net, CLI.

Hệ thống sẽ sử dụng Cloudinary để lưu trữ các hình ảnh của bất động sản.

## Pythonanywhere

## Firebase Cloud Message

# THIẾT KẾ HỆ THỐNG

## Use case

## Thiết kế cơ sở dữ liệu

Cơ sở dữ liệu được chia làm 6 nhóm chính:

* User: Quản lý dữ liệu người dùng.
* Location: Quản lý dữ liệu liên quan đế vị trí.
* Estate: Quản lý dữ liệu liên quan đến thông tin bất động sản.
* Post: Quản lý dữ liệu liên quan đến bài đăng.
* Notification: Quản lý dữ liệu liên quan đến hệ thống thông báo sự thay đổi thông tin bất động sản.
* Other: Quản lý các dữ liệu bổ sung cho hệ thống.

### Nhóm quản lý dữ liệu người dùng

Bao gồm các bảng **User**, **UserToken**. Bảng **User** phục vụ cho việc xác định và quản lý người dùng, đồng thời kết nối thông tin của người dùng với hệ thống dữ liệu bất động sản. Bảng **UserToken** phục vụ chức năng đăng nhập vào hệ thống.

**Bảng 1.** User

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| UserID | Int | Dùng để định danh người dùng |
| Name | String | Tên người dùng |
| Username | String | Thông tin đăng nhập, được khởi tạo khi người dùng đăng ký tài khoản |
| Password | String | Mật khẩu để đăng nhập |
| Gender | Text | Giới tính |
| Birthday | Date | Ngày sinh của người dùng |
| Address | String | Thông tin nơi ở của người dùng |
| Avatar | String | Ảnh đại diện, được lưu trữ dưới dạng đường dẫn đến ảnh đại diện |
| PhoneNumber | String | Thông tin số điện thoại |
| Email | String | Địa chỉ email của người dùng |
| IdentifyNumber | String | Số chứng minh thư/thẻ căn cước của người dùng |
| IsAgency | Boolean | Xác định thông tin tài khoản có được nâng cấp lên tài khoản môi giới hay chưa |
| Pin | Int | Mã Pin của tài khoản, dùng để reset mật khẩu |

**Bảng 2.** User token

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| UserTokenID | Int | Dùng để định danh token của người dùng |
| UserID | Int | Khóa ngoại đến người dùng, mục đích là xác định token của người dùng nào |
| Token | String | Đại diện cho một người dùng. Dùng để định danh người dùng khi tương tác với hệ thống |

### Nhóm quản lý dữ liệu liên quan đến vị trí

Cung cấp các thông tin liên quan đến vị trí, phục vụ cho việc tìm kiếm các thông tin về bất động sản. Thông tin vị trí được chia ra theo các cấp bậc Thành phố/Tỉnh, Quận/Huyện, Phường/Xã/Thị trấn, Đường/Phố.

Mỗi địa điểm bất động sản được định nghĩa dựa trên các bảng được xây dựng sẵn. Bao gồm các bảng **Province, District, Ward, Street**.

**Bảng 3.** Province

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| ProvinceID | Int | Dùng để định danh mỗi tỉnh/thành phố |
| Name | String | Tên của tỉnh/thành phố |
| Code | String | Mã tỉnh/thành phố |

**Bảng 4.** District

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| DistrictID | Int | Dùng để định danh mỗi quận/huyện |
| Name | String | Tên của quận/huyện |
| Prefix | String | Mã quận/huyện |
| ProvinceID | Int | Khóa ngoại được liên kết với tỉnh/thành phố mà quận/huyện trực thuộc. |

**Bảng 5.** Ward

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| WardID | Int | Dùng để định danh mỗi phường/xã |
| Name | String | Tên của phường/xã |
| Prefix | String | Mã phường/xã |
| ProvinceID | Int | Khóa ngoại được liên kết với tỉnh/thành phố mà phường/xã trực thuộc |
| DistrictID | Int | Khóa ngoại được liên kết với quận/huyện mà phường/xã trực thuộc |

**Bảng 6.** Street

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| StreetID | Int | Dùng để định danh đường/phố |
| Name | String | Tên của đường/phố |
| Prefix | String | Mã đường/phố |
| ProvinceID | Int | Khóa ngoại được liên kết với tỉnh/thành phố mà đường/phố trực thuộc |
| DistrictID | Int | Khóa ngoại được liên kết với quận/huyện mà đường/phố trực thuộc |

Trong cơ sở dữ liệu của hệ thống, thông tin địa điểm có thể sử dụng được trong toàn bộ lãnh thổ của Việt Nam.

### Nhóm quản lý dữ liệu liên quan đến thông tin bất động sản

Thông tin bất động sản quan trọng trong việc mua bán bất động sản., nhằm phục vụ quá trình đăng bài, sửa bài đăng,... Thông tin về dữ liệu bất động sản được xem là một khuôn mẫu cho mỗi bất động sản, khi đó nếu người dùng muốn đăng một bài viết, việc làm của người dùng là cần cung cấp những thông tin cần thiết cho từng loại bất động sản đó.

Để quá trình phân loại, tìm kiếm thông tin bất động sản trở nên dễ dàng hơn, chúng tôi có thêm một số thành phần khác của một bất động sản như sau:

* Bảng EstateStatus

Bảng EstateStatus cho biết rằng trạng thái của bất động sản có hết hạn hay chưa. Và được thiết kế với cấu trúc dữ liệu như sau:

**Bảng 7.** EstateStatus

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| EstateStatusID | Int | Định danh trạng thái của bất động sản |
| Status | Bool | Trạng thái của bất động sản. Nếu còn hạn giá trị là True, ngược lại là False. |

* Bảng EstateType

Định nghĩa loại bất động sản mà nó thuộc về. Dùng để phân loại bất động sản, giúp dễ dàng trong việc tìm kiếm, sắp xếp, lọc. Hiện tại, hệ thống có định nghĩa các loại bất động sản như sau: Nhà riêng; Căn hộ, chung cư; Biệt thự; Nhà mặt phố; Đất nền; Khu nghỉ dưỡng, trang trại; Nhà kho và Bất động sản khác.

**Bảng 8.** EstateType

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| EstateTypeID | Int | Dùng để xác định loại bất động sản |
| Name | String | Tên loại bất động sản |

* Bảng Project

Dùng để lưu trữ thông tin dự án của bất động sản đó. Thông tin dự án bao gồm tên, địa điểm và vị trí của dự án.

**Bảng 9.** Project

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| ProjectID | Int | Dùng để xác định một dự án |
| Name | String | Tên của dự án |
| ProvinceID | Int | Khóa ngoại liên kết với bảng tỉnh/thành phố |
| DistrictID | Int | Khóa ngoại liên kết với bảng quận/huyện |
| Lat | Float | Vĩ độ |
| Lng | Float | Kinh độ |

* Bảng Transaction

Dùng để xác định xem loại bất động sản hiện tại được đăng lên với mục đích là bán hay thuê.

**Bảng 10.** Transaction

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| TransactionID | Int | Dùng để xác định loại giao dịch |
| Name | String | Tên của loại giao dịch |

* Bảng EstateImage

Lưu trữ những hình ảnh của bất động sản. Hình ảnh được lưu dưới dạng một đường dẫn đến nơi lưu trữ của hình ảnh trên server.

**Bảng 11.** EstateImage

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| EstateImageID | Int | Định danh hình ảnh |
| EstateID | Int | Khóa ngoại đến bảng bất động sản, cho biết hình ảnh này là của bất động sản nào. |
| Image | String | Đường dẫn đến nơi lưu trữ của hình ảnh |

Sau đây là cấu trúc dữ liệu chính để quản lý thông tin bất động sản:

**Bảng 12.** Estate

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| EstateID | Int | Định danh bất động sản |
| Title | String | Nội dung tiêu đề |
| EstateTypeID | Int | Khóa ngoại được liên kết với bảng loại bất động sản, mô tả loại bất động sản |
| EstateStatusID | Int | Khóa ngoại được liên kết với bảng trạng thái bất động sản |
| ProjectID | Int | Khóa ngoại được liên kết với bảng dự án |
| ProviceID | Int | Khóa ngoại được liên kết với bảng tỉnh/thành phố |
| DistrictID | Int | Khóa ngoại được liên kết với bảng quận/huyện |
| WardID | Int | Khóa ngoại được liên kết với bảng phường/xã |
| StreetID | Int | Khóa ngoại được liên kết với bảng đường/phố |
| AddressDetail | String | Dùng để mô tả chính xác vị trí của bất động sản nếu các dữ liệu về tỉnh, phường, xã, đường chưa mô tả được chính xác. |
| NumberOfRoom | Int | Số phòng của bất động sản |
| Description | String | Thông tin mô tả sơ bộ về bất động sản |
| Detail | String | Thông tin mô tả chi tiết về bất động sản như hướng nhà, nội thất... |
| Price | Int | Giá của bất động sản |
| Area | Float | Diện tích của bất động sản |
| Contact | String | Thông tin liên hệ |
| CreatedDay | Datetime | Ngày được đăng |
| Lat | Float | Vĩ độ |
| Lng | Float | Kinh độ |
| Transaction | Int | Khóa ngoại được liên kết với bảng loại giao dịch |
| IsApproved | Bool | Cho biết bài đăng có được chấp nhận hay chưa |

### Nhóm quản lý dữ liệu liên quan đến bài đăng

Một bài đăng được quản lý bởi hai thông tin: thông tin của người dùng và thông tin của bất động sản. Giúp cho việc tìm kiếm, truy vấn dễ dàng hơn. Bên cạnh đó, việc thống kê cũng trở nên thuận tiện.

Cấu trúc dữ liệu của một bài đăng như sau:

**Bảng 13.** Post

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| PostID | Int | Thông tin định danh một bài đăng |
| UserID | Int | Khóa ngoại đến bảng thông tin người dùng. |
| EstateID | Int | Khóa ngoại đến bảng bất động sản, cho biết hình ảnh này là của bất động sản nào. |
| DateFrom | Datetime | Thời gian bài đăng được khởi tạo. |
| DateTo | Datatime | Thời gian bài đăng sẽ hết hạn. |

### Nhóm quản lý dữ liệu liên quan đến thông báo

Hệ thống có cung cấp chức năng thông báo cho người dùng. Chức năng thông báo sẽ thông báo cho người dùng nếu các bài đăng mà người dùng yêu thích có chỉnh sửa thông tin.

Chức năng thông báo có hai phần:

* Sử dụng Firebase Cloud Message (FCM) để thông báo cho người dùng khi bài đăng có chỉnh sửa trong thời gian thực.
* Thông báo trong ứng dụng. Nếu người dùng vì một lý do nào đó không xem được thông báo từ FCM, người dùng có thể xem lại tại mục thông báo trong ứng dụng.

Các cấu trúc dữ liệu như sau:

**Bảng 14.** UserNotiToken

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| ID | Int | Thông tin định danh |
| UserID | Int | Khóa ngoại đến bảng thông tin người dùng. |
| Token | String | Chứa đựng một FCM client token của người dùng. Nhằm mục đích thông báo cho firebase cloud message biết gửi một thông báo cho thiết bị nào. |

**Bảng 15.** Notification

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| ID | Int | Thông tin định danh. |
| EstateID | Int | Khóa ngoại liên kết với bảng thông tin bất động sản, mục đích để định danh thông báo này là của bất động sản nào. |
| Timestamp | Datetime | Thời gian thông báo được khởi tạo. |

**Bảng 16.** NotificationData

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| ID | Int | Thông tin định danh. |
| UserID | Int | Khóa ngoại đến bảng thông tin người dùng, để định danh thông báo được gửi cho người dùng nào. |
| NotificationID | Int | Khóa ngoại để bảng nội dung thông báo. |
| State | Boolean | Xác định xem thông báo này đã được đọc hay chưa? |

### Nhóm quản lý các dữ liệu bổ sung

Ngoài ra còn có một số cấu trúc bảng bổ sung, có nhiệm vụ cung cấp dữ liệu cho hệ thống. Ví dụ như:

* Cấu trúc bảng SortType: Định nghĩa các loại sắp xếp.
* Cấu trúc bảng FilterMaxPrice: Định nghĩa các giá trị max cho từng bộ lọc.
* Cấu trúc bảng FilterMinPrice: Định nghĩa các giá trị min cho từng bộ lọc.
* Cấu trúc bảng FilterPostTime: Định nghĩa các giá trị thời gian cho bộ lọc.
* Cấu trúc bảng FilterArea: Định nghĩa các giá trị về diện tích, phục vụ cho chức năng lọc, tìm kiếm.
* Cấu trúc bảng FilterNumberOfRoom: Định nghĩa các giá trị về số phòng, phục vụ cho chức năng lọc, tìm kiếm.

References:

* <https://restfulapi.net/>
* <https://www.mulesoft.com/resources/api/what-is-rest-api-design>
* <https://www.json.org/json-en.html>
* <https://o7planning.org/vi/11385/gioi-thieu-ve-python>
* <https://www.javatpoint.com/python-history>
* <https://www.djangoproject.com/start/overview/>
* <https://www.hostinger.vn/huong-dan/mysql-la-gi/>
* <https://en.wikipedia.org/wiki/MySQL>
* <https://github.com/cloudinary/cloudinary_gem>