ĐẠI HỌC QUỐC GIA TP.HCM

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA HỆ THỐNG THÔNG TIN**

****

BÁO CÁO

ĐỒ ÁN HỆ THỐNG THÔNG TIN

**HỆ THỐNG ĐĂNG TIN VÀ HỖ TRỢ**

**TÌM KIẾM BẤT ĐỘNG SẢN**

Giáo viên hướng dẫn**: ThS. CAO THỊ KIM TUYẾN**

Nhóm sinh viên thực hiện

1. **BẠCH SỸ ĐỨC VINH MSSV : 08520467**
2. **LÊ NGUYỄN HÀO HIỆP MSSV : 08520631**

Lớp **: HTTT03**

Khóa **: 2008 – 2013**

Khoa  **: HỆ THỐNG THÔNG TIN**

***TP. Hồ Chí Minh, tháng 5 năm 2012***

**MỤC LỤC**

[DANH SÁCH HÌNH VẼ iii](#_Toc322767342)

[DANH SÁCH BẢNG BIỂU v](#_Toc322767343)

[LỜI MỞ ĐẦU vi](#_Toc322767344)

[NHẬN XÉT CỦA GIẢNG VIÊN vii](#_Toc322767345)

[Chương 1: TỔNG QUAN 1](#_Toc322767346)

[1.1 Đặt vấn đề 1](#_Toc322767351)

[1.2 Mục tiêu đề tài 1](#_Toc322767352)

[1.3 Nội dung thực hiện 1](#_Toc322767353)

[1.4 Phương pháp thực hiện 2](#_Toc322767354)

[1.5 Sản phẩm của đề tài 2](#_Toc322767355)

[1.6 Phạm vi nghiên cứu 2](#_Toc322767356)

[1.7 Cấu trúc báo cáo 2](#_Toc322767357)

[Chương 2: CƠ SỞ LÝ THUYẾT 4](#_Toc322767358)

[2.1 Cơ sở dữ liệu không gian 4](#_Toc322767363)

[2.1.1 Geodatabase 4](#_Toc322767364)

[2.1.2 Cơ sở dữ liệu không gian theo chuẩn OGC 6](#_Toc322767365)

[2.2 Hệ quản trị CSDL PostgreSQL 7](#_Toc322767366)

[2.2.1 Giới thiệu chung 7](#_Toc322767367)

[2.2.2 Cơ sở dữ liệu trên PostgreSQL 8](#_Toc322767368)

[**2.2.2.1** **Bảng SPATIAL\_REF\_SYS và Hệ quy chiếu không gian** 8](#_Toc322767369)

[**2.2.2.2** **Bảng GEOMETRY\_COLUMNS** 8](#_Toc322767370)

[2.2.2.3.1 Geometry Type 9](#_Toc322767371)

[2.2.2.3.2 Geography Type 11](#_Toc322767372)

[2.2.2.3.4 So sánh geometry và geography 13](#_Toc322767373)

[2.2.3 Truy vấn không gian 14](#_Toc322767374)

[2.2.3.1 Nhóm hàm Accestor 14](#_Toc322767375)

[2.2.3.2 Nhóm hàm đo lường 16](#_Toc322767376)

[2.2.3.3 Nhóm hàm xác định mối quan hệ không gian 16](#_Toc322767377)

[2.3.3.4 Nhóm hàm đưa ra đối tượng hình học mới 18](#_Toc322767378)

[2.3.3.5 Nhóm hàm thay đổi đối tượng hình học 18](#_Toc322767379)

[2.2 Google Map Service 19](#_Toc322767380)

[Chương 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 24](#_Toc322767381)

[3.1 Khảo sát hiện trạng, xác định yêu cầu 24](#_Toc322767388)

[3.2 Phân tích yêu cầu 26](#_Toc322767389)

[3.2.1 Yêu cầu lưu trữ 26](#_Toc322767390)

[3.2.2 Yêu cầu phi chức năng 28](#_Toc322767391)

[3.3 Xây dựng mô hình Usecase 29](#_Toc322767392)

[3.3.1 Mô hình Usecase hệ thống 29](#_Toc322767393)

[3.3.1.1 Usecase chi tiết chức năng quản lý bất động sản 29](#_Toc322767394)

[3.3.1.2 Use case chi tiết chức năng tìm kiếm 29](#_Toc322767395)

[3.3.1.3 Usecase chi tiết chức năng quản lý tài khoản 30](#_Toc322767396)

[3.3.2 Danh sách các Actor của mô hình 30](#_Toc322767397)

[3.3.3 Danh sách hiện thực hóa Usecase 31](#_Toc322767398)

[3.3.4 Đặc tả Usecase 31](#_Toc322767399)

[3.4 Biểu đồ tuần tự (Sequence Diagram) và Biểu đồ lớp (Class Diagram) 32](#_Toc322767400)

[3.4.1 Biểu đồ tuần tự và biểu đồ lớp cho từng Usecase 32](#_Toc322767401)

[3.4.2 Sơ đồ lớp tổng quát của hệ thống 48](#_Toc322767402)

[3.5 Danh sách các lớp trong mô hình 48](#_Toc322767403)

[3.6 Biểu đồ trạng thái (State Diagram) 50](#_Toc322767404)

[3.7 Thiết kế CSDL 51](#_Toc322767405)

[3.7.1 Mô hình dữ liệu mức vật lý 51](#_Toc322767406)

[3.7.2 Mô tả các bảng 51](#_Toc322767407)

[3.7.2.6 Bảng Hình ảnh (Photo) 53](#_Toc322767408)

[3.7.3 Các lớp dữ liệu nền trong hệ thống 53](#_Toc322767409)

[3.8 Thiết kế giao diện chương trình 53](#_Toc322767410)

[3.8.1 Tổ chức 53](#_Toc322767411)

[3.8.2 Danh sách các màn hình 54](#_Toc322767412)

[Chương 4: CÀI ĐẶT THỰC NGHIỆM VÀ KIỂM TRA 58](#_Toc322767413)

[4.1 Cài đặt 58](#_Toc322767422)

[4.1.1 Môi trường phát triển hệ thống 58](#_Toc322767423)

[4.1.2 Môi trường triển khai 58](#_Toc322767424)

[4.2 Biểu đồ triển khai (Deployment Diagram) 58](#_Toc322767425)

[4.3 Kết quả đạt được 58](#_Toc322767426)

[Chương 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 60](#_Toc322767427)

[5.1 Kết luận 60](#_Toc322767433)

[5.2 Hướng phát triển 60](#_Toc322767434)

[PHỤ LỤC A – ĐẶC TẢ USECASE 61](#_Toc322767435)

# DANH SÁCH HÌNH VẼ

[Hình 1: Cấu trúc geodatabase 4](#_Toc322767436)

[Hình 2: Cấu trúc CSDL không gian theo OGC 6](#_Toc322767437)

[Hình 3: Kiểu hiển thị bản đồ Roadmap 20](#_Toc322767438)

[Hình 4: Kiểu hiển thị bản đồ Satellite 20](#_Toc322767439)

[Hình 5: Kiểu hiển thị bản đồ Hybird 21](#_Toc322767440)

[Hình 6: Kiểu hiển thị bản đồ Terrain 21](#_Toc322767441)

[Hình 7: Google Map Markers 22](#_Toc322767442)

[Hình 8: Google Map polyline 23](#_Toc322767443)

[Hình 9: Google Map Polygon 23](#_Toc322767444)

[Hình 10: Mô hình Usecase tổng quát 29](#_Toc322767445)

[Hình 11: Chi tiết usecase quản lý bất động sản 29](#_Toc322767446)

[Hình 12: Chi tiết usecase tìm kiếm 29](#_Toc322767447)

[Hình 13: Chi tiết usecase quản lý tài khoản 30](#_Toc322767448)

[Hình 14: Sơ đồ trình tự – Đăng ký 32](#_Toc322767449)

[Hình 15: Sơ đồ lớp – Đăng ký 32](#_Toc322767450)

[Hình 16: Sơ đồ trình tự – Đăng nhập 33](#_Toc322767451)

[Hình 17: Sơ đồ lớp – Đăng nhập 33](#_Toc322767452)

[Hình 18: Sơ đồ trình tự – Tương tác bản đồ 34](#_Toc322767453)

[Hình 19: Sơ đồ trình tự – Tìm kiếm dữ liệu thuộc tính 34](#_Toc322767454)

[Hình 20: Sơ đồ trình tự – Tìm kiếm không gian 35](#_Toc322767455)

[Hình 21: Sơ đồ lớp – Tìm kiếm 35](#_Toc322767456)

[Hình 22: Sơ đồ trình tự – So sánh bất động sản 36](#_Toc322767457)

[Hình 23: Sơ đồ lớp – So sánh bất động sản 36](#_Toc322767458)

[Hình 24: Sơ đồ trình tự – Chỉ dẫn đường đi 37](#_Toc322767459)

[Hình 25: Sơ đồ lớp – Chỉ dẫn đường đi 37](#_Toc322767460)

[Hình 26: Sơ đồ trình tự – Quản lý bất động sản 38](#_Toc322767461)

[Hình 27: Sơ đồ trình tự – Đăng tin 39](#_Toc322767462)

[Hình 28: Sơ đồ trình tự – Xóa tin 40](#_Toc322767463)

[Hình 29: Sơ đồ trình tự – Chỉnh sửa tin 41](#_Toc322767464)

[Hình 30: Sơ đồ lớp – Quản lý bất động sản 42](#_Toc322767465)

[Hình 31: Sơ đồ trình tự – Quản lý tài khoản 43](#_Toc322767466)

[Hình 32: Sơ đồ trình tự – Thêm tài khoản 43](#_Toc322767467)

[Hình 33: Sơ đồ trình tự – Xóa tài khoản 44](#_Toc322767468)

[Hình 34: Sơ đồ trình tự – Chỉnh sửa tài khoản 44](#_Toc322767469)

[Hình 35: Sơ đồ lớp – Quản lý tài khoản 45](#_Toc322767470)

[Hình 36: Sơ đồ trình tự – Duyệt bài 45](#_Toc322767471)

[Hình 37: Sơ đồ lớp – Duyệt bài 46](#_Toc322767472)

[Hình 38: Sơ đồ trình tự – Thay đổi mật khẩu 47](#_Toc322767473)

[Hình 39: Sơ đồ lớp – Thay đổi mật khẩu 47](#_Toc322767474)

[Hình 40: Sơ đồ lớp tổng quát của hệ thống 48](#_Toc322767475)

[Hình 41: Sơ đồ trạng thái Entity “User” 50](#_Toc322767476)

[Hình 42: Sơ đồ trạng thái Entity “RealEstate” 50](#_Toc322767477)

[Hình 43: Mô hình dữ liệu mức vật lý 51](#_Toc322767478)

[Hình 44: Sơ đồ màn hình dành cho người dùng 54](#_Toc322767479)

[Hình 45: Sơ đồ màn hình dành cho quản trị 54](#_Toc322767480)

[Hình 46: Sơ đồ triển khai 58](#_Toc322767481)

# DANH SÁCH BẢNG BIỂU

[Bảng 1: Danh sách nhóm hàm Accestor 16](#_Toc322767482)

[Bảng 2: Danh sách nhóm hàm đo lường 16](#_Toc322767483)

[Bảng 3: Danh sách nhóm hàm xác định mối quan hệ không gian 17](#_Toc322767484)

[Bảng 4: Danh sách nhóm hàm đưa ra đối tượng hình học mới 18](#_Toc322767485)

[Bảng 5: Danh sách nhóm hàm thay đổi đối tượng hình học 18](#_Toc322767486)

[Bảng 6: Kết quả khảo sát một số website bất động sản 24](#_Toc322767487)

[Bảng 7: Danh sách các actor 30](#_Toc322767488)

[Bảng 8: Danh sách hiện thực hóa Usecase 31](#_Toc322767489)

[Bảng 9: Danh sách các lớp trong hệ thống 49](#_Toc322767490)

[Bảng 10: Danh sách thuộc tính “RealEstate” 52](#_Toc322767491)

[Bảng 11: Danh sách thuộc tính “User” 52](#_Toc322767492)

[Bảng 12: Danh sách thuộc tính “District” 53](#_Toc322767493)

[Bảng 13: Danh sách thuộc tính “City” 53](#_Toc322767494)

[Bảng 14: Danh sách thuộc tính “Category” 53](#_Toc322767495)

# LỜI MỞ ĐẦU

# NHẬN XÉT CỦA GIẢNG VIÊN

...............................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

# Chương 1: TỔNG QUAN

# 

# 

# 



# Đặt vấn đề

* Trước sự biến động của nền kinh tế tài chính thế giới đã thúc đẩy thị trường bất động sản không ngừng đổi mới và phát triển. Điều đó dẫn đến việc giao dịch bất động sản ngày càng sôi nổi. Bên cạnh đó phạm vi của bất động sản ngày càng được mở rộng hơn so với chỉ chú trọng vào đất và nhà trước kia, thêm vào đó các văn phòng, căn hộ và thậm chí kể cả phòng trọ cũng được chú ý đến.
* Hiện nay, nhu cầu giao dịch bất động sản là một hoạt động diễn ra thường xuyên và được thực hiện bởi nhiều tầng lớp khác nhau trong xã hội. Tuy nhiên, phần lớn đều tập trung vào cách tìm kiếm trực tiếp thông qua các trung tâm môi giới bất động sản và báo chí, điều đó có thể gây mất thời gian cũng như lãng phí cho những người có nhu cầu. Trên hết việc đó khó có thể giúp xác định một cách khách quan vị trí của bất động sản muốn hướng đến. Mặt khác, chủ bất động sản cũng gặp nhiều khó khăn trong việc truyền tải thông tin đến thị trường.

# Mục tiêu đề tài

* Nhằm nâng cao hiệu quả và tiết kiệm thời gian trong việc giao dịch bất động bất động sản, việc ứng dụng công nghệ thông tin và internet trong việc triển khai một dịch vụ trực tuyến hỗ trợ đăng tin và tìm kiếm bất động sản được thể hiện trực quan trên bản đồ số là hết sức cần thiết.
* Đề tài tập trung xây dựng website hỗ trợ đăng tin tức và tìm kiếm các thông tin bất động sản đựa trên tương tác với bản đồ và sử dụng các ưu điểm của công nghệ GIS trong truy vấn không gian.Việc ứng dụng GIS trong việc tìm kiếm cho ta một cách thức tìm kiếm mới dựa trên các nhu cầu về mặt địa lý đối với người sử dụng khác xa so với cách thức tìm kiếm dựa trên từ khóa truyền thống.

# Nội dung thực hiện

* Nghiên cứu, tìm hiểu các thông tin cần thiết của bất động sản trong giao dịch thực tế và một số website hiện tại kết hợp với quy trình đăng tin tức trực tuyến.
* Nghiên cứu CSDL không gian theo 2 hướng tiếp cận khác nhau và quy trình triển khai một CSDL không gian trên hệ quản trị CSDL PostgreSQL.
* Phân tích, xác định các đối tượng cần quản lý. Xây dựng dữ liệu không gian lưu trữ dữ liệu bất động sản và thu thập dữ liệu mẫu.
* Xây dựng các lớp dữ liệu nền.
* Xây dựng và triển khai hệ thống đăng tin và tìm kiếm.

# Phương pháp thực hiện

* Khảo sát, thu thập thông tin, yêu cầu trong việc giao dịch bất động sản trên địa bàn thành phố Hồ Chí Minh.
* Tiếp cận nghiên cứu và kết hợp công nghệ Web và GIS.
* Phân tích thiết kế hệ thống hướng đối tượng theo ngôn ngữ mô hình hóa UML.
* Lập trình triển khai và kiểm nghiệm hệ thống.

# Sản phẩm của đề tài

* CSDL không gian với dữ liệu nền và dữ liệu chuyên đề bất động sản khu vực thành phố Hồ Chí Minh.
* Website đăng tin và hỗ trợ tìm kiếm bất động sản bao gồm các chức năng:
  + Người dùng đăng tin và tìm kiếm bằng từ khóa, tương tác bản đồ, tìm kiếm năng cao trên dữ liệu thuộc tính tìm kiếm nâng cao bằng truy vấn không gian và chỉ dẫn đường đi đến bất động sản.
  + Quản trị website đăng tin cũng như xét duyệt các tin tức do người dùng đăng để hiển thị trên website, quản lý người dùng.

# Phạm vi nghiên cứu

* **Phạm vi địa lý**: Một số quận trung tâm khu vực thành phố Hồ Chí Minh. Dữ liệu thu thập và dữ liệu nền chỉ mang tính chất tham khảo, không có mục đích giao dịch, quảng bá hoặc chỉ định đến bất cứ tổ chức hoặc cá nhân nào.
* **Phạm vi công nghệ**: Sử dụng công nghệ mã nguồn mở: Google Map Service để xây dựng WebGIS với bản đồ số, Hệ quản trị cơ sở dữ liệu PostgrSQL, Ngôn ngữ lập trình web PHP với Framework CodeIgniter.

# Cấu trúc báo cáo

Báo cáo có trang bao gồm:

* **Phần nội dung chính**

Chương 1: Trình bày tổng quan đề tài và các vấn đề liên quan khi thực hiện đề tài.

Chương 2: Trình bày cơ sở lý thuyết và các vấn đề công nghệ, kĩ thuật được nghiên cứu và áp dụng trong đề tài.

Chương 3: Trình bày chi tiết quá trình phân tích và thiết kế hệ thống dựa trên ngôn ngữ mô hình hóa UML.

Chương 4: Trình bày chi tiết cách thức triển khai và vận hành hệ thống kết hợp với đánh giá các thành quả đã đạt được.

Chương 5: Trình bày tổng kết chung về đề tài và đưa ra các hướng phát triển của đề tài trong tương lai.

* **Phần phụ lục**

# Chương 2: CƠ SỞ LÝ THUYẾT

# 



# Cơ sở dữ liệu không gian

# Geodatabase

* Geodatabase là một mô hình dữ liệu không gian do ESRI đưa ra dùng cho việc lưu trữ, truy vấn và xử lý dữ liệu GIS.
* Geodatabase là mô hình dữ liệu hướng đối tượng, cho phép lưu trữ thống nhất dữ liệu không gian và phi không gian khi ánh xạ mô hình đã thiết kế xuống hệ quản trị cơ sở dữ liệu, hỗ trợ nhiều người truy cập dữ liệu đồng thời tại một thời điểm. Trong mô hình này, các thực thể được mô tả như các đối tượng với các thuộc tính, hành động và các quan hệ. Geodatabase hỗ trợ được sự phức tạp của các loại đối tượng địa lý khác nhau, cho phép định nghĩa các quan hệ giữa các đối tượng với các luật cho việc duy trì tính ràng buộc toàn vẹn giữa chúng.
* Dữ liệu trong Geodatabase được tổ chức thành các lớp dữ liệu dạng vector hoặc raster. Dữ liệu Vector được thể hiện một trong 3 dạng: điểm, đường hoặc là vùng và có thể được gắn hệ tọa độ gọi là tham chiếu không gian.
* Hiện tại Geodatabase có thể được cài đặt lên hầu hết các hệ quản trị cơ sở dữ liệu như: Oracle, SQL server, MS Access, Postgres, DB2.

**2.1.1.1 Cấu trúc Geodatabase**



Hình : Cấu trúc geodatabase

* **Objects classes**
  + Một đối tượng (Object) là một thể hiện của một lớp đối tượng.
  + Tất cả các đối tượng(Objects) một lớp đối tượng(object class) có cùng đặc tính và hành vi.
  + Các đối tượng liên quan với nhau được thể hiện qua các mối quan hệ.
* **Feature classes**
  + Tập hợp các thể hiện của đối tượng không gian, các đối tượng kế thừa các đặc tính của Object
  + Có hình dạng hình học và vị trí xác định trên cùng một hệ quy chiếu không gian
  + Có các thuộc tính và hành vi cụ thể
* **Feature datasets**
  + Tập các feature class hay object class nằm trong cùng một hệ tọa độ.Feature dataset tương đương với một bản đồ
* **Domains**
  + Domains: Chỉ rõ những giá trị có thể chấp nhận được của của một trường (field).Có 2 loại domain: Miền giá trị ( Range), Mã hóa các giá trị(Code values).
  + Domains có thể dùng chung qua nhiều lớp đối tượng.
* **Relationship classes**
  + Relationship classes: lưu trữ mối quan hệ của các đối tượng trong feature class. Geodatabase duy trì tính toàn vẹn tham chiếu của mối quan hệ khi đối tượng được xóa.

**2.1.1.2 Các đối tượng không gian trong Geodatabase**

* **Điểm(Point)**

Biểu diễn các đối tượng không gian không có miền bao quanh hay độ dài, nhiều khi nó còn biểu diễn các đối tượng không gian có kích thước quá nhỏ so với tỷ lệ bản đồ. Ví dụ: cột móc biên giới, đồng hồ nước.

* **Đường (PolyLines)**

Biểu diễn các đối tượng không gian có chiều dài xác định nhưng không có miền bao hay những đối tượng không gian tỷ lệ rất hẹp so với tỷ lệ bản đồ. Ví dụ : Đường ống nước, biên giới, đường giao thông, sông hồ…

* **Vùng(Polygon)**

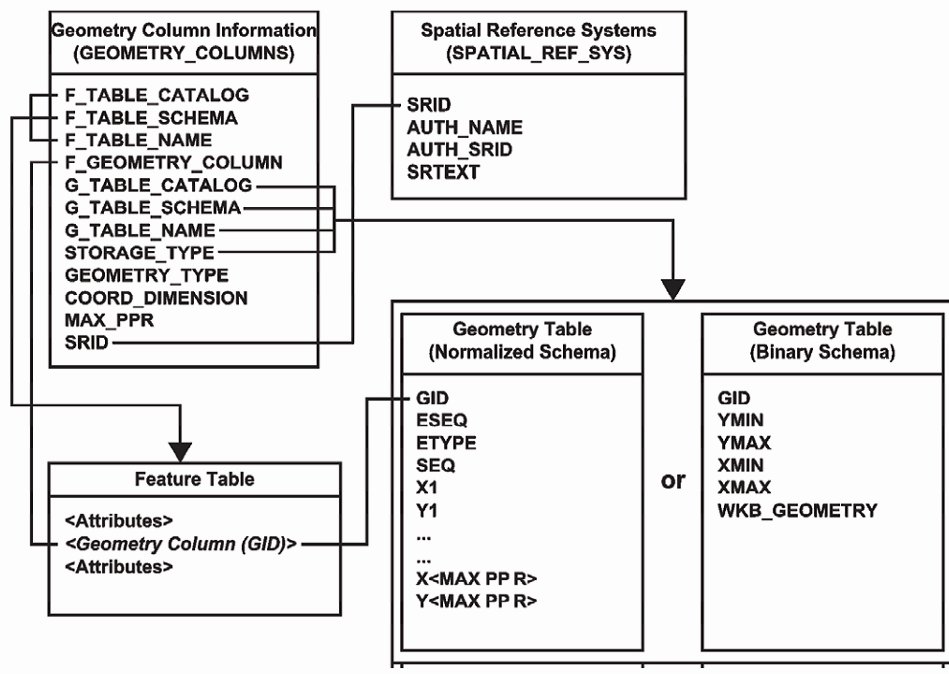
Biểu diễn các đối tượng có miền bao xác định và khép kín. Ví dụ: ruộng, ao hồ, các đơn vị hành chính

**2.1.1.3 Tính Planar Topology trong Geodatabase**

* Topology trong geodatabase cho phép chúng ta biểu diễn hình học dùng chung cạnh giữa các đối tượng trong một lớp đối tượng và giữa các lớp đối tượng khác nhau.
* Có thể tổ chức các đối tượng trong một geodatabase để tạo planar topologies. Feature classes có thể chia sẻ cạnh, nút với các lớp đối tượng khác trong planar topology, planar topology tạo thành từ các điểm nút, cạnh và bề mặt.

# Cơ sở dữ liệu không gian theo chuẩn OGC

* Giản đồ sau giới thiệu về cấu trúc bảng và mối liên hệ giữa các bảng thuộc tính, giữa bảng lưu trữ thông tin hình học với nhau.



Hình : Cấu trúc CSDL không gian theo OGC

* Bảng **GEOMETRY\_COLUMNS** mô tả các đặc tính hình học nhằm quản lý kiểu và hệ thống tọa độ của đối tượng hình học đó.
* Bảng **SPATIAL\_REF\_SYS** mô tả hệ tọa độ và những phép biến đổi tọa độ.
* Bảng **FEATURE TABLE** chứa các thuộc tính mô tả đặc trưng của một đối tượng nào đó. Kiểu hình học của một đối tượng là một thuộc tính trong bảng và được tham chiếu đến bảng **GEOMETRY\_COLUMN** và bảng **GEOMETRY TABLE**.
* Bảng hình học (Geometry Table) lưu trữ những đối tượng hình học sử dụng kiểu số hoặc nhị phân.
* Dựa vào kiểu lưu trữ được xác định bởi bảng **GEOMETRY\_COLUMNS**, một đối tượng hình học được lưu trữ như một mảng những giá trị toạ độ (Normalized Schema) hoặc như một giá trị nhị phân đơn lẻ (Binary Schema). Trong những trường hợp trước đây, những kiểu dữ liệu số của SQL được dùng cho những toạ độ và những giá trị số này được chứa trong bảng hình học (geometry table) chođến khi đối tượng hình học được chuẩn hoá.

# Hệ quản trị CSDL PostgreSQL

# Giới thiệu chung

* PostgreSQL là hệ thống quản trị cơ sở dữ liệu quan hệ đối tượng dựa trên POSTGRES bản 4.2, được phát triển tại trường đại học California tại phòng nghiên cứu máy tính Berkeley. Nó là một chương trình mã nguồn mở xây dựng trên mã nguồn ban đầu của đại học Berkeley. Nó hỗ trợ một phần rất lớn cho SQL chuẩn và cung cấp nhiều tính năng hiện đại như:
* Các truy vấn phức tạp
* Khóa ngoại
* Trigger
* Khung nhìn
* Tính toàn vẹn của các giao dịch
* Kiểm tra truy cập đồng thời đa phiên bản.
* Ngoài ra, PostgreSQL có thể được mở rộng bởi nhiều người dùng bằng nhiều cách, ví dụ, người dùng có thể thêm kiểu dữ liệu, hàm, toán tử, hàm tập hợp, phương thức đánh chỉ mục và ngôn ngữ thủ tục.
* PostgreSQL cũng là hệ quản trị cơ sở dữ liệu hỗ trợ mạnh trong việc lưu trữ dữ liệu không gian. PostgreSQL kết hợp với module PostGIS cho phép người dùng lưu trữ các lớp dữ liệu không gian.
* Tính tương thích , độ bảo mật cao, ngày càng được hoàn thiện nhờ cộng đồng người sử dụng đóng góp và phát triển.
* PostgreSQL được phổ biến bằng giấy phép BSD cổ điển. Nó không quy định những hạn chế trong việc sử dụng mã nguồn của phần mềm. Bởi vậy PostgreSQL có thể được dùng, sửa đổi và phổ biến bởi bất kỳ ai cho bất kỳ mục đích nào.

# Cơ sở dữ liệu trên PostgreSQL

* + - 1. **Bảng SPATIAL\_REF\_SYS và Hệ quy chiếu không gian**
* Chứa các thông số về các hệ quy chiếu không gian tham chiếu cho các đối tượng trong một spatial database.
* Cấu trúc bảng SPATIAL\_REF\_SYS như sau

CREATE TABLE spatial\_ref\_sys(

srid INTEGER NOT NULL PRIMARY KEY ,

auth\_name VARCHAR (256),

auth\_srid INTEGER,

srtext VARCHAR (2048),

proj4text VARCHAR (2048)

)

* Cột SRID là định danh duy nhất (có thể hiểu như khóa chính của bảng dữ liệu).
* CộtAUTH\_NAME mô tả cơ quan hoặc tổ chức định nghĩa và sử dụng hệ thống tham chiếu.
* Cột AUTH\_SRID là số nguyên được gán bởi cơ quan hoặc tổ chức, còn cột SRTEXT hiển thị WKT của hệ thống tham chiếu không gian.
* **Lưu ý**: Mặc định, bảng SPATAIL\_REF\_SYS được lấy từ hệ thống cơ sở dữ liệu tham chiếu không gian ESPG. Ví dụ như số SRID trong bảng SPATAIL\_REF\_TABLE và định danh ESPG luôn luôn giống nhau.
  + - 1. **Bảng GEOMETRY\_COLUMNS**
* Cấu trúc bảng như sau

CREATE TABLE geometry\_columns(

f\_table\_catalog VARRCHAR(256) NOT NULL ,

f\_table\_schema VARCHAR(256) NOT NULL ,

f\_table\_name VARCHAR(256) NOT NULL ,

f\_geometry\_column VARCHAR(256) NOT NULL ,

coord\_dimension INTEGER NOT NULL ,

srid INTEGER NOT NULL ,

type VARCHAR(30) NOT NULL

)

* Mỗi cột không gian được xác định duy nhất bởi sự kết hợp của shema/table/column. Cột COORD\_DIMENSION xác định chiều không gian (2, 3 hoặc 4 chiều) của cột. Cột SRID mô tả hệ thống tham chiếu không gian, nó là khóa ngoài tham chiếu đến bảng SPATAIL\_REF\_SYS. Cuối cùng là cột cột TYPE mô tả kiểu hình học được mô tả trongbảng.

**2.2.2.3 Biểu diễn dữ liệu không gian**

* Ngoài các dữ liệu thông thường, Postgres hỗ trợ lưu trữ dữ liệu không gian với 2 định dạng chính là GEOMETRY và GEOGRAPHY.

# Geometry Type

* Các đối tượng không gian được PostGIS hỗ trợ được mở rộng từ chuẩn ISO 19125–Simple Feature của OGC. Bên cạnh việc hỗ trợ tất cả các kiểu đối tượng và hàm được định nghĩa với chuẩn ISO 19125, PostGIS còn mở rộng việc hỗ trợ các tọa độ 3DZ, 3DM và 4D.
* Các chuẩn OpenGIS định nghĩa hai cách cho biểu diễn dữ liệu không gian: WKT và WBT. Cả WKT và WBT đều bao gồm thông tin kiểu và tọa độ không gian của đối tượng.

Ví dụ về biểu diễn đối tượng không gian với WKT:

• POINT(0 0)

• LINESTRING(0 0,1 1,1 2)

• POLYGON((0 0,4 0,4 4,0 4,0 0),(1 1, 2 1, 2 2, 1 2,1 1))

• MULTIPOINT(0 0,1 2)

• MULTILINESTRING((0 0,1 1,1 2),(2 3,3 2,5 4))

• MULTIPOLYGON(((0 0,4 0,4 4,0 4,0 0),(1 1,2 1,2 2,1 2,1 1)), ((–1 –1,–1 –2,–2 –2,–2 –1,–1 –1)))

• GEOMETRYCOLLECTION(POINT(2 3),LINESTRING(2 3,3 4))

Các chuẩn OpenGIS cũng yêu cầu các định dạng lưu trữ các đối tượng không gian phải chứa đựng thông tin về hệ quy chiếu không gian, được thể hiện thông qua tham số SRID.Tham số SRID thường bắt buộc khi tạo các đối tượng không gian để insert vào các cơ sở dữ liệu không gian.

PostGIS hỗ trợ việc nhập xuất các đối tượng không gian dưới dạng WKT, WKB thông qua các hàm sau:

bytea WKB = ST\_AsBinary(geometry);

text WKT = ST\_AsText(geometry);

geometry = ST\_GeomFromWKB(bytea WKB, SRID);

geometry = ST\_GeometryFromText(text WKT, SRID);

* **PostGIS EWKB, EWKT và Canonical Forms**
* WKT, WKB có một số hạn chế như chỉ hỗ trợ đối tượng 2D, không hỗ trợ biểu diễn tham số SRID gán với đối tượng. Vì lý do đó, PostGIS cũng hỗ trợ hai dạng mở rộng của WKT, WKB là EWKT và EWKB (Lưu ý: hai định dạng này chưa phải là chuẩn chính thức của OGC, nên có thể không được hỗ trợ trong tương lai).
* Các định dạng EWKB/EWKT hỗ trợ biểu diễn tọa độ 3DM, 3DZ, 4D và cho phép nhúng thông tin về SRID khi biểu diễn đối tượng.

Ví dụ một số cách biểu diễn đối tượng không gian với EWKT:

• POINT(0 0 0) –– XYZ

• SRID=32632;POINT(0 0) –– XY cùng với SRID

• POINTM(0 0 0) –– XYM

• POINT(0 0 0 0) –– XYZM

• SRID=4326;MULTIPOINTM(0 0 0,1 2 1) –– XYM cùng với SRID

• MULTILINESTRING((0 0 0,1 1 0,1 2 1),(2 3 1,3 2 1,5 4 1))

• POLYGON((0 0 0,4 0 0,4 4 0,0 4 0,0 0 0),(1 1 0,2 1 0,2 2 0,1 2 0,1 1 0))

• MULTIPOLYGON(((0 0 0,4 0 0,4 4 0,0 4 0,0 0 0),(1 1 0,2 1 0,2 2 0,1 2 0,1 1 0)),((–1 –1 0,–1 –2 0,–2 –2 0,–2 –1 0,–1 –1 0)))

• GEOMETRYCOLLECTIONM(POINTM(2 3 9), LINESTRINGM(2 3 4, 3 4 5))

PostGIS hỗ trợ việc nhập xuất các đối tượng không gian dưới dạng WKT, WKB thông qua các hàm sau:

bytea EWKB = ST\_AsEWKB(geometry);

text EWKT = ST\_AsEWKT(geometry);

geometry = ST\_GeomFromEWKB(bytea EWKB);

geometry = ST\_GeomFromEWKT(text EWKT);

* **SQL–MM Part 3**
* Chuẩn SQL Multimedia Applications Spatial(SQL–MM) mở rộng chuẩn ISO 19125 bằng cách định nghĩa 1 số phương pháp nội suy đường cong.
* Định nghĩa SQL–MM bao gồm tọa độ 3dm, 3dz and 4d, nhưng ko cho phép nhúng thông tin SRID
* WKT chưa hỗ trợ đầy đủ với SQL–MM. Các đối tượng dạng cung được biểu diễn như sau:

• CIRCULARSTRING(0 0, 1 1, 1 0)

CIRCULARSTRING(0 0, 4 0, 4 4, 0 4, 0 0)

CIRCULARSTRING là loại đường cong cơ bản. Một đoạn đơn đòi hỏi ba điểm, bắt đầu, kết thúc và điểm còn lại nằm đâu đó trên cung. Ngoại lệ là đường cong khép kín, đầu cuối trùng nhau. Trong trường hợp này, điểm còn lại phải là trung tâm của cung, như là phía đối diện của vòng tròn. Để kết các cung lại với nhau, điểm cuối của cung này sẽ là điểm đầu của cung kia, tương tự như LINESTRING. Điều đó có nghĩa vòng tròn chuỗi phải có số lẻ các điểm lớn hơn 1.

• COMPOUNDCURVE(CIRCULARSTRING(0 0, 1 1, 1 0),(1 0, 0 1))

* Một đường cong ghép là 1 cung đơn giản, cung liên tục có cả đoạn cong và đoạn thẳng. điều này có nghĩa rằng để thêm những thành phần chính xác, điểm cuối của mỗi thành phần phải trùng khớp với điểm bắt đầu của thành phần kế tiếp.

• CURVEPOLYGON(CIRCULARSTRING(0 0, 4 0, 4 4, 0 4, 0 0),(1 1, 3 3, 3 1, 1 1))

* Ví dụ đường cong ghép trong một polygon

CURVEPOLYGON(COMPOUNDCURVE(CIRCULARSTRING(0 0,2 0, 2 1,

2 3, 4 3),(4 3, 4 5, 1 4, 0 0)), CIRCULARSTRING(1.7 1, 1.4 0.4, 1.6 0.4, 1.6 0.5, 1.7 1) )

* Một CURVEPOLYGON giống một polygon, với 1 vòng tròn ở phía ngoài và nhiều vòng hơn ở phía trong. Sự khác nhau là 1 vòng có thể nhận một hình dạng của một vòng tròn chuỗi, chuỗi đoạn thẳng hoặc là đường cong ghép. PostGIS 1.4 PostGIS hỗ trợ đường cong ghép và polygon.

• MULTICURVE((0 0, 5 5),CIRCULARSTRING(4 0, 4 4, 8 4))

* Một MULTICURVE là một tập các cung, có chứa chuỗi đoạn thẳng, chuỗi đường cong hoặc chuỗi ghép.

• MULTISURFACE(CURVEPOLYGON(CIRCULARSTRING(0 0, 4 0, 4 4, 0 4, 0 0),(1 1, 3 3, 3 1, 1 1)),((10 10, 14 12, 11 10,10 10),(11 11, 11.5 11, 11 11.5, 11 11)))

* Tập các vùng, có thể là polygon thẳng hoặc polygon cong.

# Geography Type

* Kiểu GEOGRAPHY hỗ trợ việc biểu diễn, tính toán cho các đối tượng không gian theo tọa độ địa lý (hay kinh/vị độ).
* Để phân biệt rõ hơn khác biệt giữa kiểu Geography và kiểu Geometry, ta xét đến ví dụ tính toán đường bay ngắn nhất của hai điểm trên bản đồ. Với kiểu Geometry, vốn dựa trên hệ tọa độ De Carte, khoảng cách này chính là độ dài của đoạn thẳng nối hai điểm với nhau. Trong khi đó với kiểu Geography, việc tính toán dựa trên một khối cầu (sphere), do đó khoảng cách này là một cung tròn lớn trên khối cầu này.
* Việc tính toán trên các đối tượng kiểu Geography thường phức tạp hơn rất nhiều so với các đối tượng Geometry. Do đó, số lượng các hàm được định nghĩa dành cho kiểu Geography vẫn chưa nhiều bằng kiểu Geometry.
* Một hạn chế khác của kiểu Geography là chỉ hỗ trợ hệ quy chiếu WGS84 (SRID: 4326).
* Trong PostGIS, kiểu dữ liệu dành cho các đối tượng GeoGraphy là geography. Trừ các đối tượng dạng cung (CURVE), PostGIS hỗ trợ đầy đủ các định dạng OGC chuẩn cho Geography.
* **Các kiểu Geography cơ bản**
* Kiểu Geography chỉ hỗ trợ những phần đơn giản nhất của chuẩn ISO 19125. Các đối tượng Geometry chuẩn sẽ được tự động ép kiểu thành Geography nếu các đối tượng đó sử dụng hệ quy chiếu WGS 84.
* Ngoài ra, ta có thể dùng các định dạng EWKT và EWKB chèn dữ liệu Geography vào cơ sở dữ liệu
* Ví dụ:

CREATE TABLE testgeog(gid serial PRIMARY KEY, the\_geog geography(POINT,4326) );

CREATE TABLE testgeog(gid serial PRIMARY KEY, the\_geog geography(POINTZ,4326) );

* Ngoài POINT và POINTZ, PostGIS còn hỗ trợ Geography với các loại đối tượng khác:

• LINESTRING

• POLYGON

• MULTIPOINT

• MULTILINESTRING

• MULTIPOLYGON

• GEOMETRYCOLLECTION

* Các cột Geography không được đăng ký trong bảng geometry\_columns, mà sẽ được đăng ký ở bảng geography\_columns. Đồng thời các kiểu này cũng được tạo trực tiếp khi tạo bảng chứ không cần dùng hàm như **AddGeometryColumn().** Ví dụ:

CREATE TABLE global\_points (

id SERIAL PRIMARY KEY,

name VARCHAR(64),

location GEOGRAPHY(POINT,4326)

);

* Tham số thứ nhất chỉ định dạng đối tượng, tham số thứ hai chỉ định hệ quy chiếu. Như đã đề cập ở trên, hiện tại kiểu Geography mới chỉ hỗ trợ hệ quy chiếu WGS84, nhưng trong tương lại việc hỗ trợ các hệ quy chiếu khác sẽ được bổ sung.
* Trong tương lai, chuyển đổi hệ tham chiếu không gian sẽ cho phép tính toán trên tọa độ cầu khác WGS84.

# So sánh geometry và geography

* Kiểu GEOGRAPHY cho phép bạn lưu dữ liệu trên hệ tọa độ kinh, vĩ độ, nhưng có một số nhược điểm sẽ có ít hàm định nghĩa trên đối tượng Geography so với Geometry; những hàm này đồng thời sẽ tốn thời gian chạy của CPU hơn.
* Việc lựa chọn sử dụng kiểu Geography hay Geometry thường tùy thuộc và nhu cầu ứng dụng và phạm vi dữ liệu không gian của ứng dụng.
* Nếu chỉ sử dụng dữ liệu không gian của một khu vực nhỏ, chúng ta có thể tìm một phép chiếu thích hợp và sử dụng kiểu Geometry, nhờ đó tận dụng được các nhiều hàm tính toán có sẵn và nâng cao hiệu năng.
* Nếu ứng dụng sử dụng dữ liệu không gian của toàn trái đất, hoặc của một vùng lục địa, sử dụng kiểu Geography cho phép xây dựng hệ thống mà không cần quan tâm chi tiết đến các hệ tham chiếu, chỉ cần lưu trữ dữ liệu với kinh/vĩ độ, và sử dụng các hàm được định nghĩa trên kiểu Geography.
* Trong trường hợp không hiểu rõ về các hệ tham chiếu, việc dùng kiểu Geography sẽ dễ dàng hơn so với kiểu Geometry.

# Truy vấn không gian

# 2.2.3.1 Nhóm hàm Accestor

|  |  |  |
| --- | --- | --- |
| **Cú pháp** | **Kết quả trả về** | **Ý nghĩa** |
| ST\_AsText(geometry) | Text | Trả về dạng text của đối tượng, Ví dụ: LINESTRING(0 0,1 1) |
| ST\_AsBinary(geometry) | WKB | Trả về dạng WKB theo OGC |
| ST\_SRID(geometry) | Int | Trả về số nguyên sẽ tham chiếu đến hệ thống không gian hình học. |
| ST\_Dimension(geometry) | Int | Trả về kích thước của đối tượng hình học .Trả về 0 nếu là điểm, 1 nếu là đường,2 là polygon |
| ST\_Envelope(geometry) | Geometry | trả về một đối tượng hình học hợp lệ((POINT, LINESTRING or POLYGON)) |
| ST\_IsEmpty(geometry) | Int | Trả về 1 nếu geometry là rỗng |
| ST\_IsSimple(geometry) | Int | Trả về 1 nếu đối tượng hình học không có các giao điểm đặc biệt. |
| ST\_IsClosed(geometry) | Int | Trả về 1 nếu đối tượng hình học bắt đầu và kết thúc trùng nhau. |
| ST\_IsRing(geometry) | Int | Trả về 1 nếu đường câu khép kính điểm bắt đầu và kết thúc trùng nhau, và đương cong không đi qua một điểm quá 2 lần. |
| ST\_NumGeometries(geometry) | Int | Nếu đối tượng hình học là tập hợp các đối tượng hình học thì trả về số lượng các đối tượng hình học. Ngược lại trả về null |
| ST\_GeometryN(geometry) | Int | Trả về N’th nếu đối tượng hình học là GEOMETRYCOLLECTION, MULTIPOINT, MULTILINESTRING hoặc MULTIPOLYGON .Ngược lại Trả về NULL. |
| ST\_NumPoints(geometry) | Int | Tìm và trả vế số điểm trong linestring đầu tiên trong đối tượng hình học.Nếu đối tượng hình học không có linestring Trả về NULL. |
| ST\_PointN(geometry,interger) | Int | trả về N’th point trong linestring đầu tiên.Trả về NULL nếu không có linestring. |
| ST\_ExteriorRing(geometry) | Geometry | Trả về vòng nằm ngoài đối tượng polygon. Trả về null nếu geometry không phải là polygon |
| ST\_NumInteriorRings(geometry) | Int | Trả về số vòng nằm ngoài polygon đầu tiên của geometry. Trả về null nếu không cò polygon |
| ST\_EndPoint(geometry) | Geometry | Trả về point cuối cùng của linestring |
| ST\_StartPoint(geometry) | Geometry | Trả về điểm đầu tiên của Linestring. |
| GeometryType(geometry) | Text | trả về tên hình học của geometry VD: 'LINESTRING', 'POLYGON', 'MULTIPOINT', |
| ST\_X(geometry) | Float | Trả về tọa độ X của điểm |
| ST\_Y(geometry) | Float | Trả về tọa độ Y của điểm |
| ST\_Z(geometry) | Float | Trả về tọa độ Z của điểm.Trả về null nếu điểm không có tọa độ |
| ST\_M(geometry) | Float | Trả về tọa độ M của điểm .Trả về null nếu không có. |

Bảng : Danh sách nhóm hàm Accestor

# 2.2.3.2 Nhóm hàm đo lường

|  |  |  |
| --- | --- | --- |
| **Cú pháp** | **Kết quả trả về** | **Ý nghĩa** |
| ST\_Distance(geometry,geometry) | Float | Tính khoảng cách giữa 2 đối tượng không gian: điểm–điểm, điểm–đường. Đơn vị mặc định là mét |
| ST\_Length(geometry) | Float | Độ dài của hình 2D nếu chúng là LINESTRING hoặc MULTILINESTRING. Đơn vị mặc định là mét |
| ST\_Area(geometry) | Float | Diện tích của POLYGON hoặc MULTIPOLIGON, mặc định đơn vị là mét vuông |
| ST\_Perimeter(geometry) | Float | Chu vi của POLYGON hoặc MULTIPOLIGON, mặc định đơn vị là mét. |

Bảng : Danh sách nhóm hàm đo lường

# Nhóm hàm xác định mối quan hệ không gian

|  |  |  |
| --- | --- | --- |
| **Cú pháp** | **Kết quả trả về** | **Ý nghĩa** |
| ST\_DWithin(geometry,geometry,float) | Boolean | Trả về true nếu đối tượng hình học này cách đối tượng hình học kia một khoảng cách quy định |
| ST\_Equals(geometry,geometry) | Boolean | Trả về true nếu 2 đối tượng không gian bằng nhau |
| ST\_Disjoint(geometry,geometry) | Boolean | Trả về true nếu 2 đối tượng không giao nhau |
| ST\_intersets(geometry,geometry) | Boolean | Trả về true nếu đối tượng giao nhau |
| ST\_touches(geometry,geometry) | Boolean | Trả về true nếu 2 đối tượng hình học có ít nhất một điểm chung, Chú ý không dùm hàm này cho Point–Point |
| ST\_Crosses(geometry,geometry) | Boolean | Trả về true 2 đối tượng giao nhau , vùng giao nhau của 2 đối tượng , phải có số chiều nhỏ hơn max số chiều của hai đối tượng nhập vào |
| ST\_Within(geometry A, geometry B) | Boolean | Trả về true nếu A hoàn toàn nằm bên trong B |
| ST\_Overlaps(geometry,geometry) | Boolean | Trả về true nếu hai đối tượng giao nhau nhưng không hoàn toàn chứa nhau. Chú ý một điểm trên một đường thẳng không phải là overlap |
| ST\_Contains(geometry A, geometry B) | Boolean | Trả về true nếu A chứa B |
| ST\_Covers(geometry A, geometry B) | Boolean | Trả về true nếu không có điểm nào của A nằm ngoài B |
| ST\_Intersects(geometry, geometry) | Boolean | Trả về true nếu 2 đối tượng giao nhau |
| ST\_Relate(geometry, geometry, intersectionPatternMatrix) | Boolean |  |
| ST\_Relate(geometry, geometry) | Boolean |  |

Bảng : Danh sách nhóm hàm xác định mối quan hệ không gian

# 2.3.3.4 Nhóm hàm đưa ra đối tượng hình học mới

|  |  |  |
| --- | --- | --- |
| **Cú pháp** | **Kết quả trả về** | **Ý nghĩa** |
| ST\_Intersection(geometry A, geometry B) | geometry | Trả về một hình, hiển thị phần chung giữa hình A và hình B. Nếu hình A và hình B không có bất kỳ điểm chung thì trả về đối tượng hình rỗng |
| ST\_Difference(geometry geomA, geometry geomB) | geometry | Trả về một hình hiển thị phần của hình A mà không giao với hình B |
| ST\_Union (geometry g1, geometry g2) | geometry | Trả về kết quả là tập hợp tất cả các hình hoặc hình đơn lẻ (MULTI) |
| ST\_SymDifference (geometry geomA, geometry geomB) | geometry | trả về một hình hiển thị phần của hình A vàhình B không giao nhau |

Bảng : Danh sách nhóm hàm đưa ra đối tượng hình học mới

# 2.3.3.5 Nhóm hàm thay đổi đối tượng hình học

|  |  |  |
| --- | --- | --- |
| **Cú pháp** | **Kết quả trả về** | **Ý nghĩa** |
| ST\_Buffer (geometry g1, float R)  ST\_Buffer(geometry g1, float R, integer num\_seg\_quater\_circle)  ST\_Buffer(geometry g1, float R, text buffer\_style\_parameters)  ST\_Buffer(geography g1, float radius\_of\_buffer\_in\_meters) | geometry | Chức năng của hàm ST\_Buffer trả về môt hình hiển thị cho tất cả các điểm mà khoảng cách của chúng từ hình <= khoảng cách.  Tùy chọn buffer\_style :  *+ quad\_segs:* số đoạn được sử dụng để xấp xỉ ¼ vòng tròn (mặc định là 8).  *+ endcap= round|flat|square:* kiểu kết thúc, mặc định là *round.*  *+ joint=round|mitre|bevel:* kiểu nối, mặc định là *round*  *+ mitre\_limit :* tỉ lệ giới hạn mép  Đơn vị của bán kính được đo bằng đơn vị của hệ thống tham chiếu không gian. |

Bảng : Danh sách nhóm hàm thay đổi đối tượng hình học

**Ví dụ**

|  |  |
| --- | --- |
| SELECT ST\_Buffer(ST\_GeomFromText(‘POINT(100, 90)’),50, ‘quad\_segs=8’); | SELECT ST\_Buffer (ST\_GeomFromText(‘POINT(100,90)’), 50,‘quad\_seds=2’); |
| SELECT ST\_Buffer(ST\_GeomFromText(‘LINESTRING(50 50,150 150, 150 50)’), 10, ‘endcap=round join=round’); | SELECT ST\_Buffer(ST\_GeomFromText(‘LINESTRING(50 50,150 150, 150 50)’), 10, ‘endcap=square join=round’); |

# Google Map Service

* Google Maps là một dịch vụ của Google, cung cấp các ứng dụng và cộng nghệ hoàn toàn miễn phí về dịch vụ bản đồ trên web, bao gồm Google Maps website (website về Google Maps), Google Ride Finder (cho phép người lái xe xác định được vị trí và hướng, cách thức để đi đến đích), các bản đồ nhúng trên các website thông qua Google Map API…
* Google Maps cung cấp cho người dùng nhiều tính năng như: tìm kiếm địa điểm thông qua GeoCoding, tìm đường đi giữa các địa điểm cùng với hành trình tương ứng. Người dùng có thể dễ dàng tiếp cận thông qua tương tác trực tiếp trên bản đồ số bằng các thao tác click, scroll…
* Google Map API được cung cấp cho các nhà phát triển có thể nhúng toàn bộ bản đồ số vào các website và thậm chí cả trên nền tảng di động. Việc sử dụng google map API không quá phức tạp thông qua một số kiến thức về đối tượng, sự kiện, điều khiển, lớp bản đồ. Phiên bản mới nhất là Google Map API V3 (3.8).

**2.2.1 Đối tượng (Map Object)**

* Phần tử cơ bản nhất trong Google Map API là các đối tượng bản đồ (Map Object) thông qua đối tượng Map. Map là một lớp trong Javascript đại diện cho các đối tượng bản đồ trên website.
* Map Object được cung cấp các loại (map type) khác nhau bao gồm
* **Roadmap** – Kiểu hiển thị bản đồ mặc định, kiểu hình vẽ bản đồ.



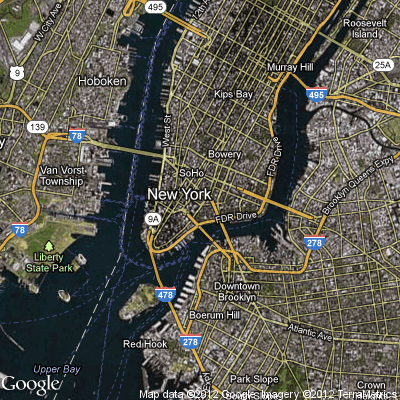
Hình : Kiểu hiển thị bản đồ Roadmap

* **Satellite** – Kiểu hiển thị ảnh vệ tinh.



Hình : Kiểu hiển thị bản đồ Satellite

* **Hybird** – Kiểu kết hợp giữa hình vẽ và ảnh vệ tinh.



Hình : Kiểu hiển thị bản đồ Hybird

* **Terrain**



Hình : Kiểu hiển thị bản đồ Terrain

**2.2.2 Sự kiện (Event)**

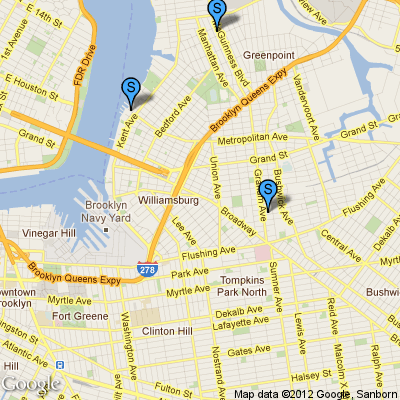
* Để bắt các sự kiện từ người dùng, Google cung cấp một đối tượng Event Listeners có tác dụng lắng nghe và bắt sự kiện. Mỗi Map Object có một số lượng các sự kiện nhất định như clicks, dbclick, move…
* Những sự kiện này hoàn toàn tách rời và độc lập so với các sự kiện trong DOM.

**2.2.3 Điều khiển (Control)**

* Google Map API cung cấp một số điều khiển (control) trong bản đồ
* RotateControlOptions: Xoay chiều bản đồ
* PanControlOptions:
* ZoomControlOptions: Thiết lập chế độ zoom.
* ScaleControlOptions: Co giãn bản đồ theo tỉ lệ.
* MapTypeControlOptions: Chuyển đổi các loại bản đồ (Terrain, Satellite….)

**2.2.4 Lớp bản đồ (Map Overlays)**

* Overlay là các đối tượng trên bản đồ được liên kết với kinh độ và tọa độ của hệ tọa độ, các đối tượng có thể di chuyển khi ta dịch chuyển bản đồ hoặc zoom bản đồ. Google Map API cung cấp các kiểu Overlay cơ bản sau
* **Marker**: Thể hiện các điểm trên bản đồ với các biểu tượng sẵn có



Hình : Google Map Markers

* **Polyline**: Thể hiện các đoạn thẳng trên bản đồ (tập hợp các điểm).



Hình : Google Map polyline

* **Polygon**: Thể hiện các vùng trên bản đồ bao gồm các tập đoạn thẳng khép kín và có hình dáng bất kỳ.

# 

Hình : Google Map Polygon

# Chương 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG



# Khảo sát hiện trạng, xác định yêu cầu

* Hiện trạng cung cấp thông tin
  + Truyền miệng: Nguồn cung cấp thông tin chính, tuy nhiên nội dung thông tin không được truyền tải chính xác và khó tìm kiếm thông tin. Khách hàng phải tiếp xúc trực tiếp với chủ bất động sản để lấy thông tin.
  + Sách – báo chí: Nguồn cung cấp thông tin tin cậy và chính xác được đa số khách hàng hiện nay tin dùng.
  + Truyền hình: Là kênh tin tức tin cậy nhưng do chi phí quảng cáo có hạn nên thông tin không được truyền tải đầy đủ và thông thường chỉ một số thông tin quan trọng.
  + Trung tâm môi giới: Nguồn tin có độ chính xác cao, tuy nhiên khách hàng thường mất một số khoản “tiền cò” giành cho “cò đất”. Hơn nữa, việc lừa đảo vẫn xảy ra trên địa bàn TPHCM bởi một số trung tâm kém chất lượng.
  + Internet: Nguồn thông tin phong phú và dễ dàng tiếp cận bất chấp mọi điều kiện về địa lý, tuy nhiên nguồn tin chưa được chuẩn hóa và chọn lọc và đôi khi gây khó khăn cho khách hàng nắm bắt thông tin.
* Khảo sát một số website hỗ trợ đăng tin tức bất động sản cho bảng kết quả sau đây.

|  |  |  |  |
| --- | --- | --- | --- |
| **Website** | **Chú giải** | **Đơn vị chủ quản** | **Hạn chế** |
| **www.batdongsan.com.vn** | Website tìm kiếm thông tin giao dịch bất động sản | Công ty cổ phần bất động sản B.D.S | Chưa hỗ trợ tìm kiếm không gian. Thông tin trình bày chưa rõ ràng. |
| [**www.sacomreal–s.com.vn**](http://www.sacomreal-s.com.vn) | Sàn giao dịch bất động sản | Ngân hàng Sacombank | Chưa hỗ trợ tìm kiếm không gian. |
| **www.map.diaoconline.vn** | Website tìm kiếm thông tin bất động sản | Công ty Cổ phần Địa Ốc Trực Tuyến | Chỉ hỗ trợ hiển thị trên bản đồ, chưa hỗ trợ tìm kiếm không gian nâng cao. |
| **www.reeland.com.vn** | Sàn giao dịch bất động sản | Công ty cổ phần bất động sản REE | Chưa hỗ trợ tìm kiếm không gian |
| **www.bandonhadat.vn** | Website tìm kiếm thông tin bất động sản | NhaVietGroup | Chỉ hỗ trợ hiển thị trên bản đồ, chưa hỗ trợ tìm kiếm không gian nâng cao. |

Bảng : Kết quả khảo sát một số website bất động sản

Hệ thống đăng tin và tìm kiếm bất động sản gồm các chứng năng chính sau.

* **Đăng tin rao bán, cho thuê bất động sản**
* Hỗ trợ người dùng đăng các tin rao bán hoặc cho thuê bất động sản:
  + Địa chỉ bất động sản.
  + Giá cả , đơn vị tính.
  + Diện tích.
  + Tình trạng pháp lý.
  + Loại hình bất động sản.
  + Loại giao dịch.
  + Thông tin chi tiết về cơ sở vật chất, tiện nghi, v.v...
* Quản lý danh mục hình ảnh liên quan đến mẩu tin rao bán, cho thuê BĐS.
* Quản lý các mẫu tin rao bán, cho thuê bất động sản.
* Thống kê truy cập, đánh giá đối với mẩu tin rao bán, cho thuê.
* Xét duyệt các tin rao bán, cho thuê từ người dùng.
* **Tìm kiếm thông tin rao bán, cho thuê bất động sản**
* Tìm kiếm thông tin BĐS dựa trên dữ liệu thuộc tính:
  + Giá
  + Loại giao dịch
  + Diện tích
  + Tình trạng pháp lý
  + Loại hình bất động sản
  + Khu vực hành chánh
  + Cơ sở vật chất, tiện nghi BĐS.
* Tìm kiếm thông qua tương tác trực tiếp với bản đồ.
* Tìm kiếm thông tin BĐS sử dụng truy vấn không gian:
  + Nằm cách đường trong phạm vi bao nhiêu mét.
  + Gần với trường học/ bệnh viện/ siêu thị/ v.v...
  + Thuộc khu vực có mật độ dân số cao/thấp.
* **So sánh, đánh giá các thông tin rao bán, cho thuê bất động sản**
* Tạo giỏ hàng.
* Lập bảng so sánh.
* Đánh giá mẩu tin BĐS dựa trên một số nhóm tiêu chí:
  + Giá cả.
  + Diện tích.
  + Tiện nghi.
  + V.v...
* **Chỉ dẫn đường đi đến vị trí bất động sản**
* **Quản lý tài khoản**

# Phân tích yêu cầu

# Yêu cầu lưu trữ

* Hệ thống cần lưu trữ những thông tin sau.
  + Thông tin người dùng
  + Thông tin bất động sản
  + Thông tin hình ảnh
  + Thông tin loại bất động sản
  + Thông tin quận/huyện
  + Thông tin thành phố/tỉnh
    - 1. **Thông tin người dùng**
* Hệ thống cần bảo mật trong tính lưu trữ và thay đổi thông tin. Vì vậy, hệ thống cung cấp tài khoản đăng nhập cho người dùng và quản trị để thực hiện các chức năng liên quan. Thông tin người dùng bao gồm.
  + Username: định danh đăng nhập.
  + Password: Mật khẩu truy cập.
  + Email: Địa chỉ Email.
  + Name: Họ tên
  + Gender: Giới tính.
  + Tel: Điện thoại.
  + IDNumber: Số CMND.
  + Add: Địa chỉ
  + Birthday: Ngày sinh.
  + Duty: Quyền hạn, ở đây có 2 quyền hạn chính là : người dùng (user) và quản trị (admin)
    - 1. **Thông tin bất động sản**
* Lưu trữ thông tin bất động sản bao gồm:
  + RealEstateID: Mã bất động sản.
  + Tittle: Tiêu đề của bất động sản.
  + Address: Địa chỉ của bất động sản.
  + Price: Giá của bất động sản.
  + Currency: Đơn vị tiền tệ
  + Unit: Đơn vị tính (tháng, m2,…)
  + Area: Diện tích.
  + Size: Kích thước.
  + Direction: Phương hướng.
  + Alley: Đường hẻm.
  + LegalStatus: Tình trạng pháp lý.
  + CategoryID: Mã loại bất động sản.
  + Transaction: Loại giao dịch (thuê, bán).
  + Description: Thông tin mô tả chi tiết về cơ sở vật chất, tiện nghi, phòng ngủ…
  + DistrictID: Quận/huyện trực thuộc.
  + Date: Ngày đăng.
  + DateModified: Ngày chỉnh sửa lần cuối.
  + AdminID: Thông tin quản trị duyệt bài lần cuối.
  + UserID: Chủ sở hữu của bất động sản.
  + Reason: Lý do không được duyệt bài
  + Status: Trạng thái của bất động sản.
  + ContactName : Tên người liên hệ.
  + ContactTel: Điện thoại liên hệ.
  + ContactAdd: Địa chỉ liên hệ.
  + Remarks: Ghi chú.
    - 1. **Thông tin hình ảnh**
* Lưu trữ thông tin về các hình ảnh liên quan của bất động sản bao gồm:
  + PhotoID: Mã hình ảnh
  + URL: đường dẫn lưu trữ của hình ảnh.
    - 1. **Thông tin loại bất động sản**
* Lưu trữ thông tin loại bất động sản bao gồm:
  + CategoryID: Mã của loại bất động sản.
  + Name: Tên của loại bất động sản.
    - 1. **Thông tin quận/huyện**
* Lưu trữ thông tin quận/bao gồm:
  + DistrictID: Mã quận/huyện
  + Name: Tên quận/huyện.
  + CityID: Mã tỉnh/thành phố trực thuộc.
    - 1. **Thông tin thành phố/tỉnh**
* Lưu trữ thông tin thành phố/tỉnh bao gồm:
  + CityID: Mã tỉnh/thành phố.
  + Name: Tên tỉnh/thành phố.

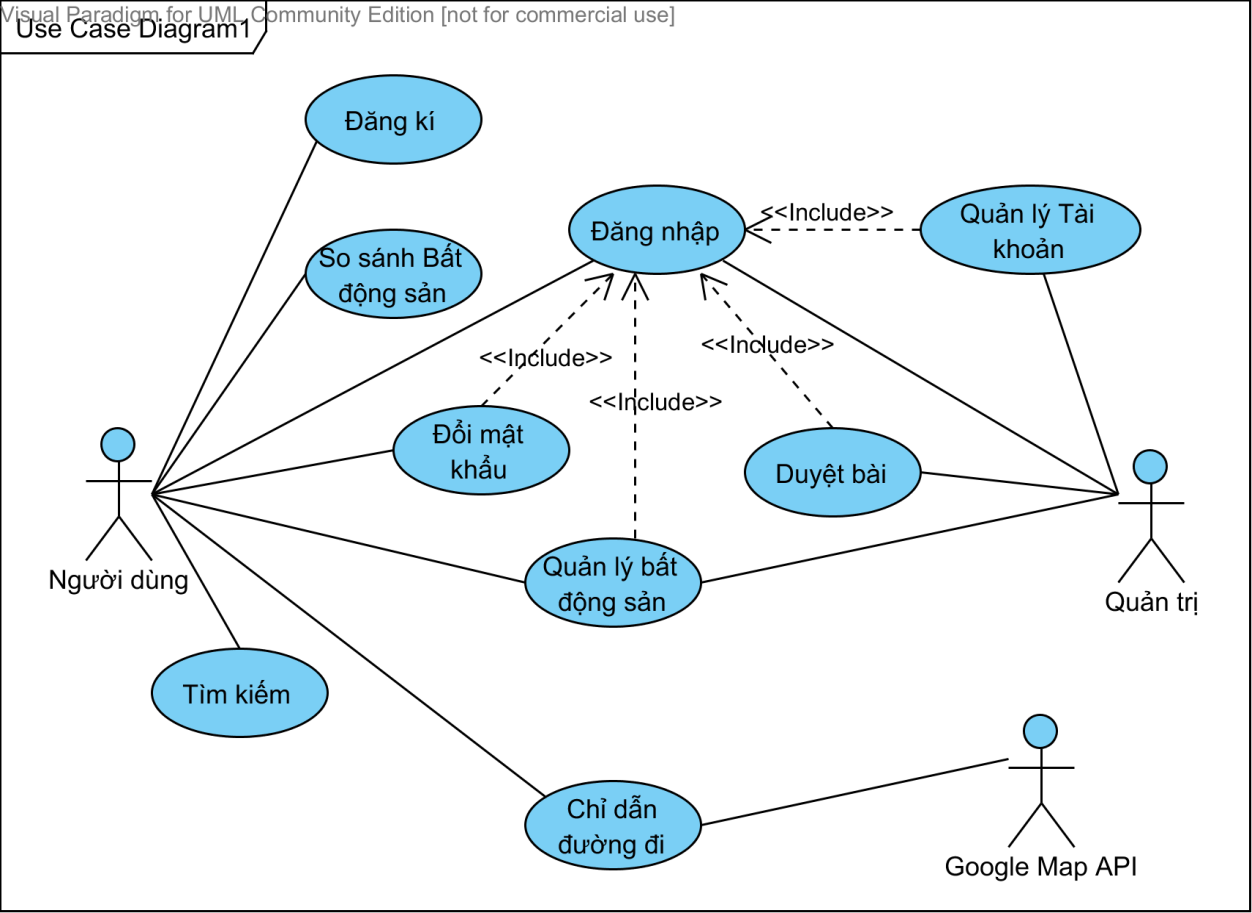
# Yêu cầu phi chức năng

* Đối với website
  + Thời gian hiển thị và phản hồi yêu cầu nhanh chóng.
  + Giao diện thân thiện với người sử dụng, từ ngữ dễ hiểu.
  + Hệ thống đảm bảo tính bảo mật, phân quyền người dùng và mã hóa mật khẩu.
* Đối với Web server
  + Cài đặt Linux hoặc Windows.
  + Hệ quản trị PostgreSQL với phần mở rộng PostGIS.
  + Cài đặt Apache, PHP, WampServer.
  + Cấu hình máy tương đối.

# Xây dựng mô hình Usecase

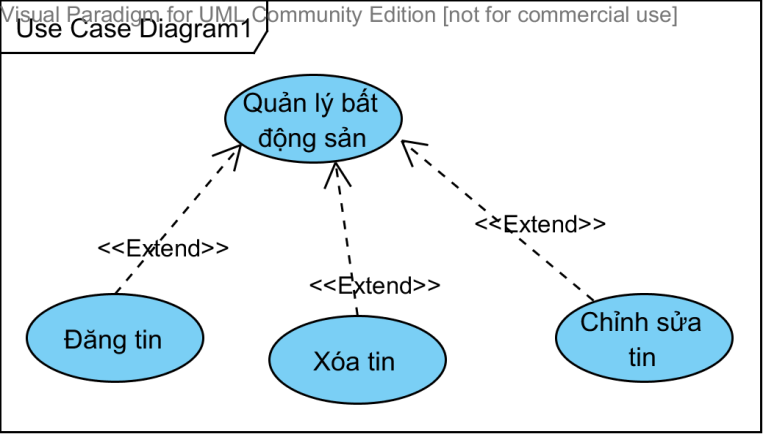
# Mô hình Usecase hệ thống

Các tác nhân tương tác với hệ thống là quản trị, người dùng và Google Map Service.



Hình : Mô hình Usecase tổng quát

# Usecase chi tiết chức năng quản lý bất động sản



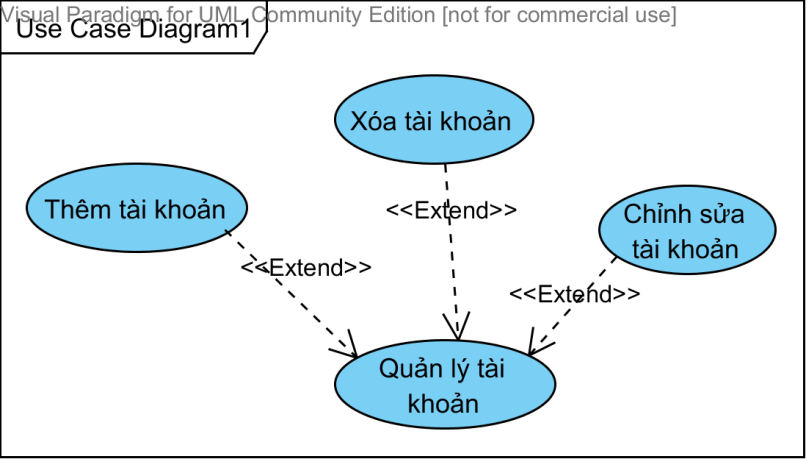
Hình : Chi tiết usecase quản lý bất động sản

# Use case chi tiết chức năng tìm kiếm



Hình : Chi tiết usecase tìm kiếm

# Usecase chi tiết chức năng quản lý tài khoản



Hình : Chi tiết usecase quản lý tài khoản

# Danh sách các Actor của mô hình

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên actor** | **Ý nghĩa/Ghi chú** |
| 1 | Người dùng | Người dùng |
| 2 | Quản trị | Quản trị của hệ thống |
| 3 | Google Map Service | Dịch vụ bản đồ số Google Map |

Bảng : Danh sách các actor

**Các usecase liên quan**

* Usecase chung cho các Actor
  + Usecase Đăng nhập
  + Usecase Quản lý bất động sản
* Actor người dùng
  + Usecase Đăng ký
  + Usecase So sánh bất động sản
  + Usecase Đổi mật khẩu
  + Usecase Tìm kiếm
* Actor quản trị
  + Usecase Duyệt bài
  + Usecase Quản lý người dùng
* Actor Google Map Service
  + Usecase chỉ dẫn đường đi

# Danh sách hiện thực hóa Usecase

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Usecase** | **Ý nghĩa/Ghi chú** |
| 1 | Đăng ký | Đăng ký thành viên để đăng nhập vào hệ thống |
| 2 | Đăng nhập | Đăng nhập vào hệ thống |
| 3 | Tìm kiếm | Tìm kiếm bất động sản dựa vào tương tác bản đồ, tìm kiếm nâng cao trên dữ liệu thuộc tính, tìm kiếm không gian |
| 4 | So sánh bất động sản | So sánh các bất động sản trên một số tiêu chí |
| 5 | Chỉ dẫn đường đi | Chỉ dẫn đường đi từ vị trí người dùng đến bất động sản |
| 6 | Quản lý bất động sản | Đăng tin, xóa tin và chỉnh sửa tin tức bất động sản. |
| 7 | Quản lý tài khoản | Thêm, xóa, chỉnh sửa tài khoản của hệ thống. |
| 8 | Duyệt bài | Duyệt để đăng tin bất động sản lên website |
| 9 | Thay đổi mật khẩu | Thay đổi mật khẩu truy cập |

Bảng : Danh sách hiện thực hóa Usecase

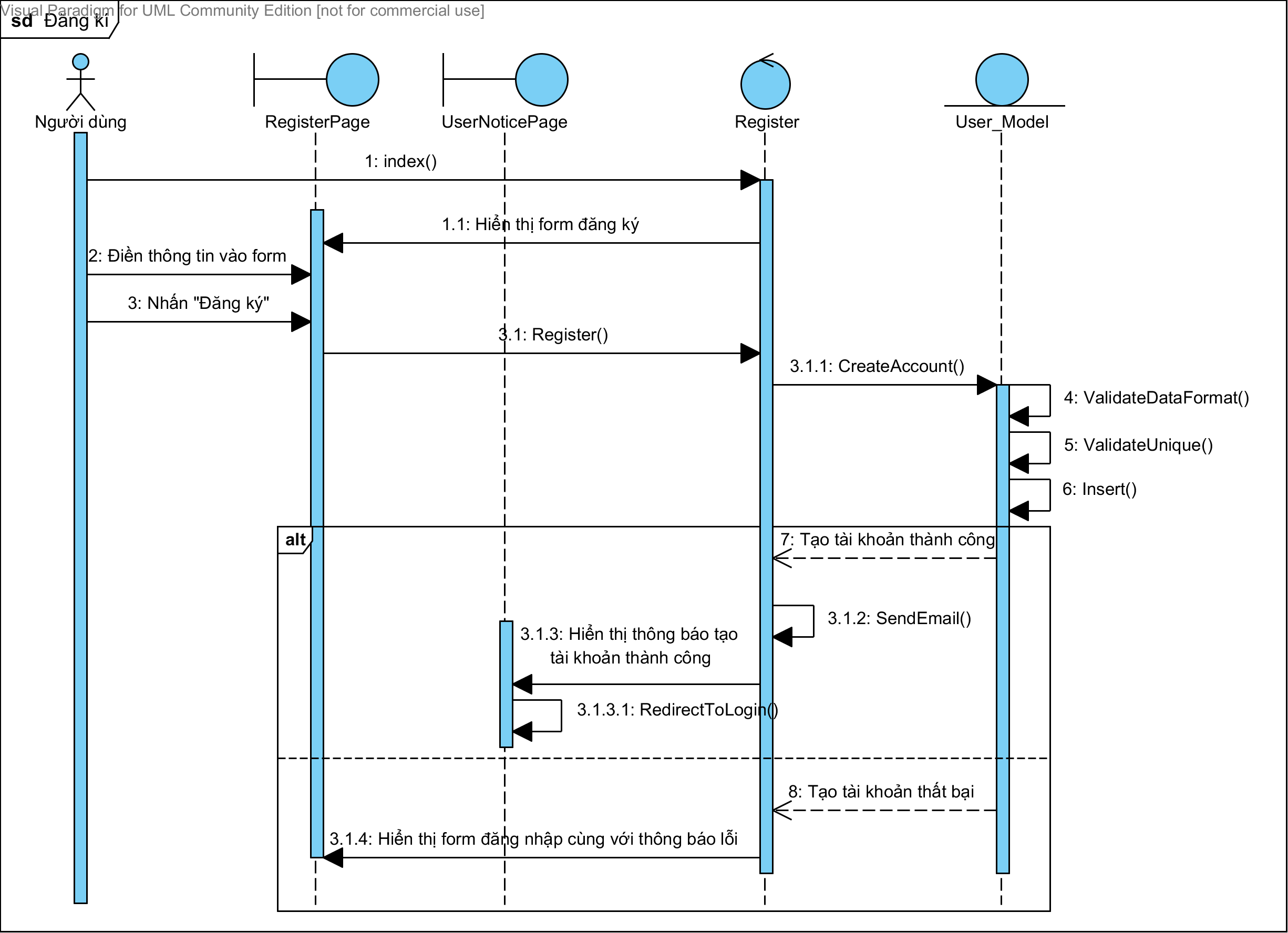
# Đặc tả Usecase

***(Xem phụ lục A – Đặc tả Usecase)***

# Biểu đồ tuần tự (Sequence Diagram) và Biểu đồ lớp (Class Diagram)

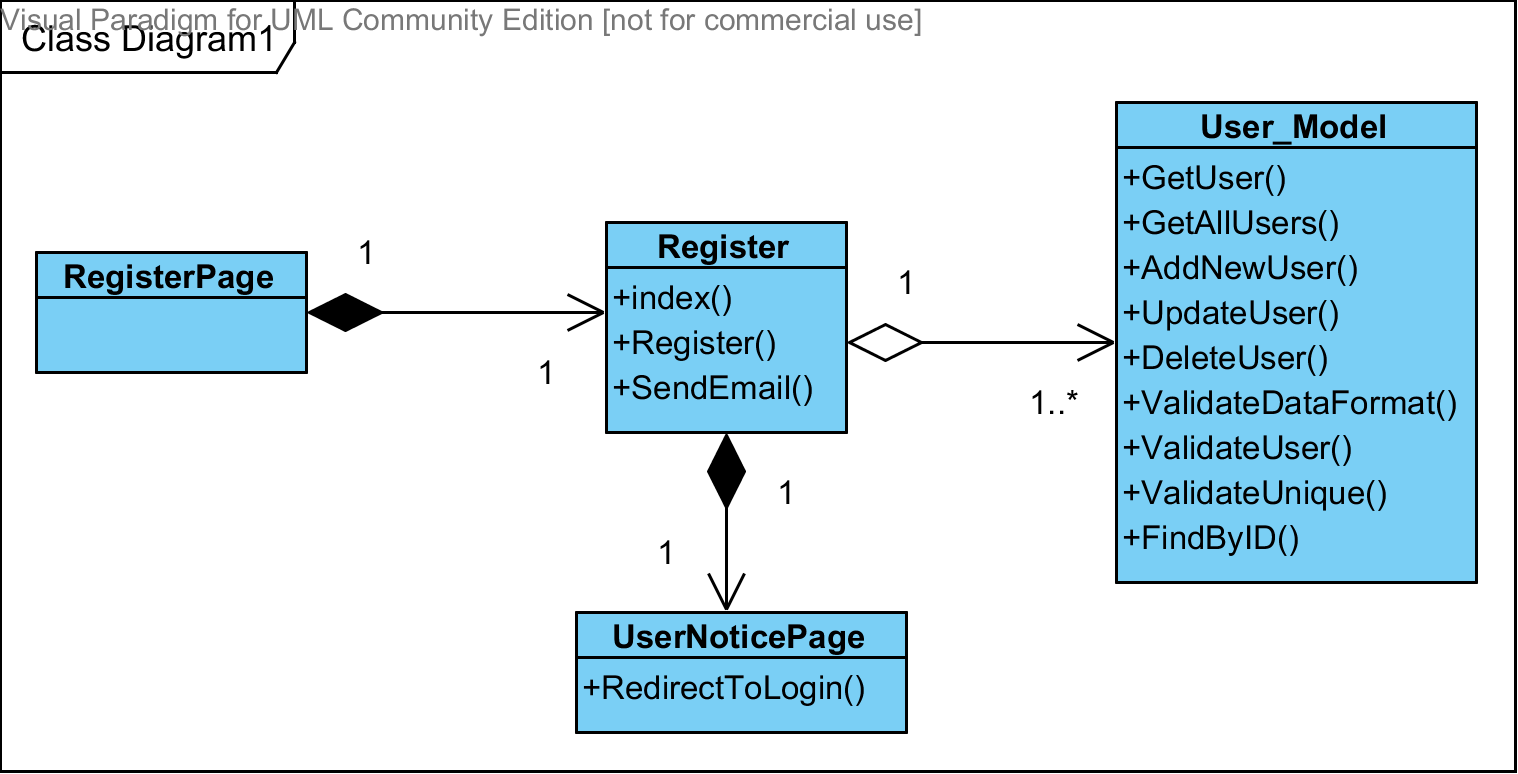
# Biểu đồ tuần tự và biểu đồ lớp cho từng Usecase

* + - 1. **Usecase Đăng ký**
         1. **Đăng ký – Basic Flow**

****

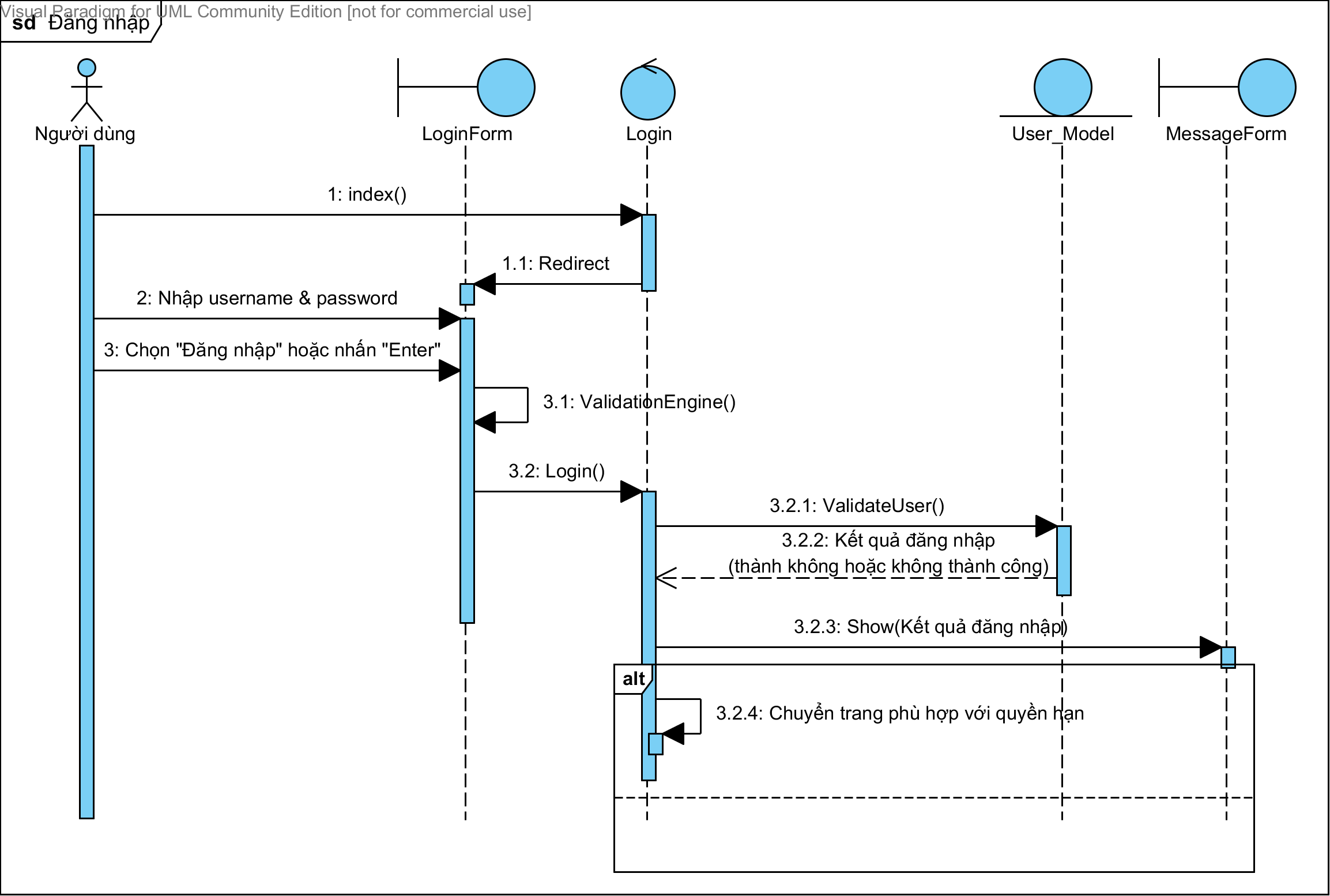
Hình : Sơ đồ trình tự – Đăng ký

* + - * 1. **Đăng ký – VOPC**

****

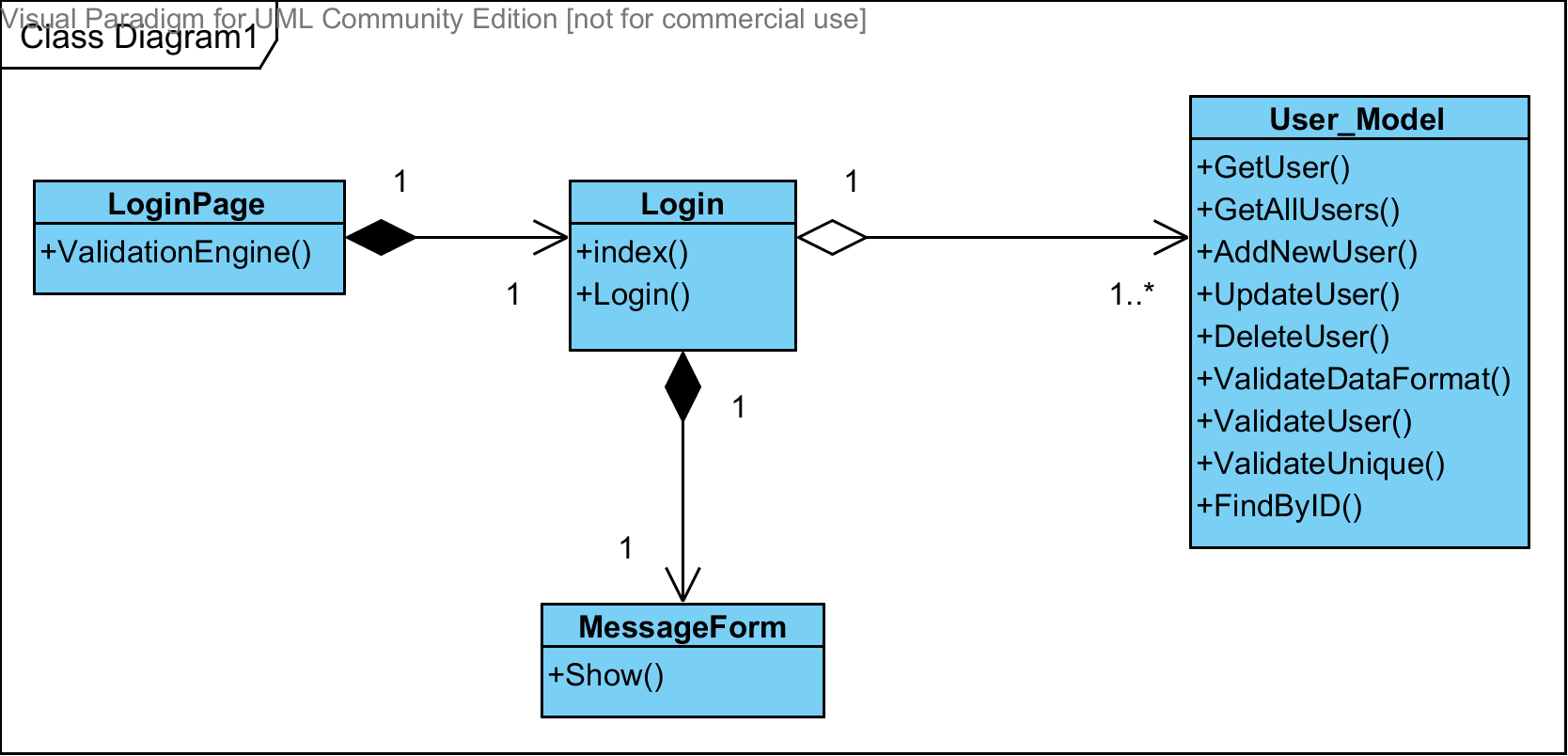
Hình : Sơ đồ lớp – Đăng ký

* + - 1. **Usecase Đăng nhập**
         1. **Đăng nhập – Basic Flow**

****

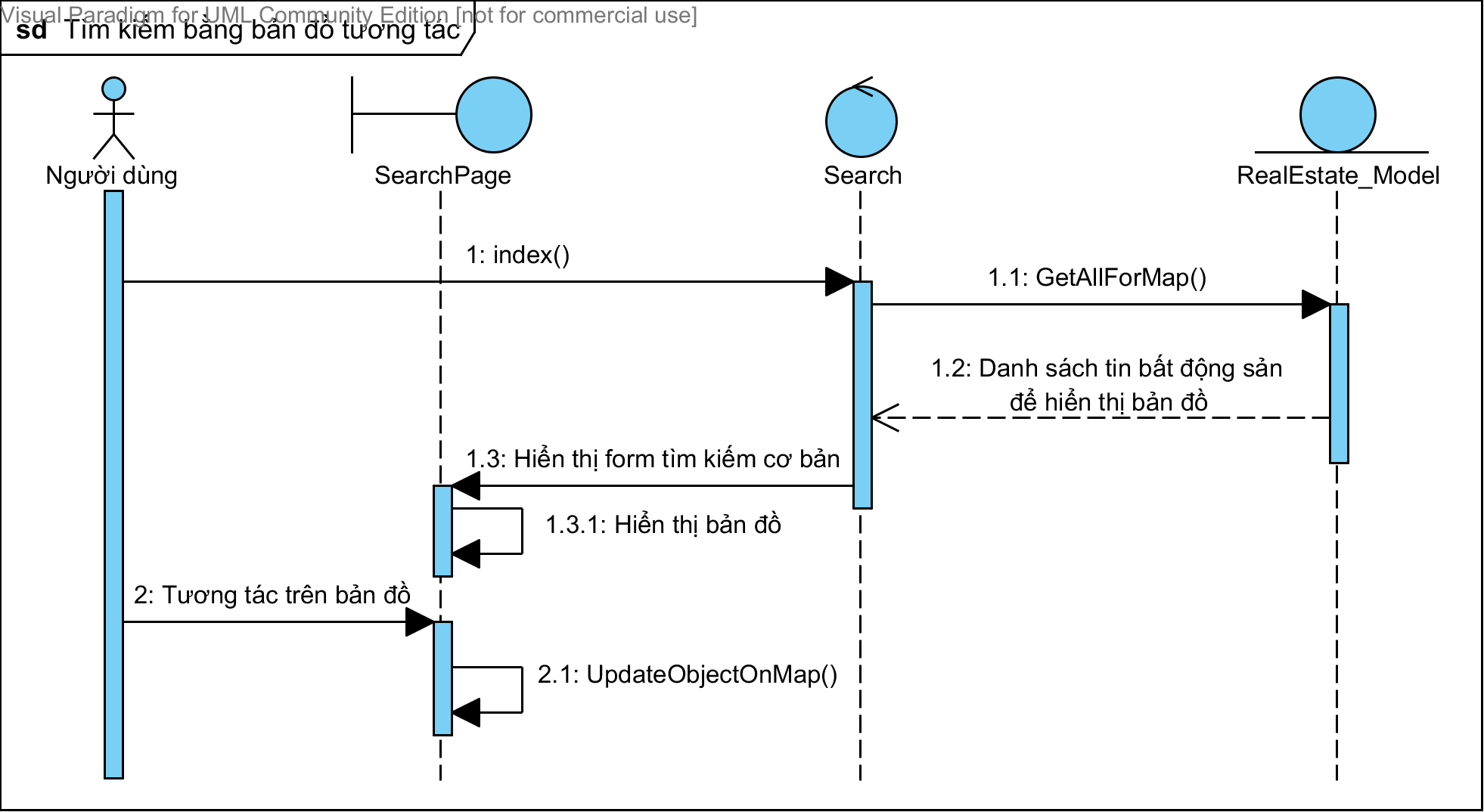
Hình : Sơ đồ trình tự – Đăng nhập

* + - * 1. **Đăng nhập – VOPC**

****

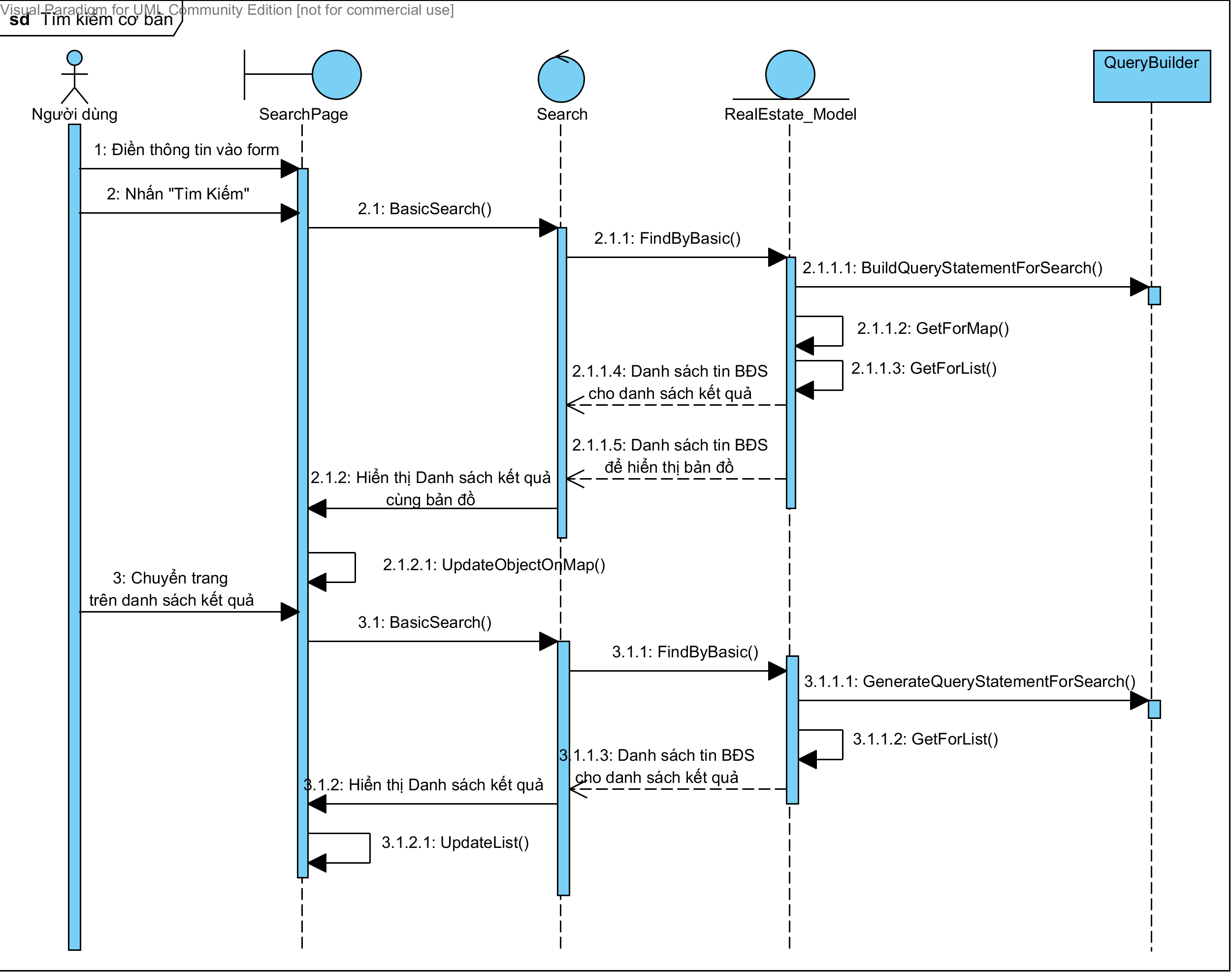
Hình : Sơ đồ lớp – Đăng nhập

* + - 1. **Usecase Tìm kiếm**
         1. **Tìm kiếm – Basic Flow**
* **Tìm kiếm tương tác bản đồ**

****

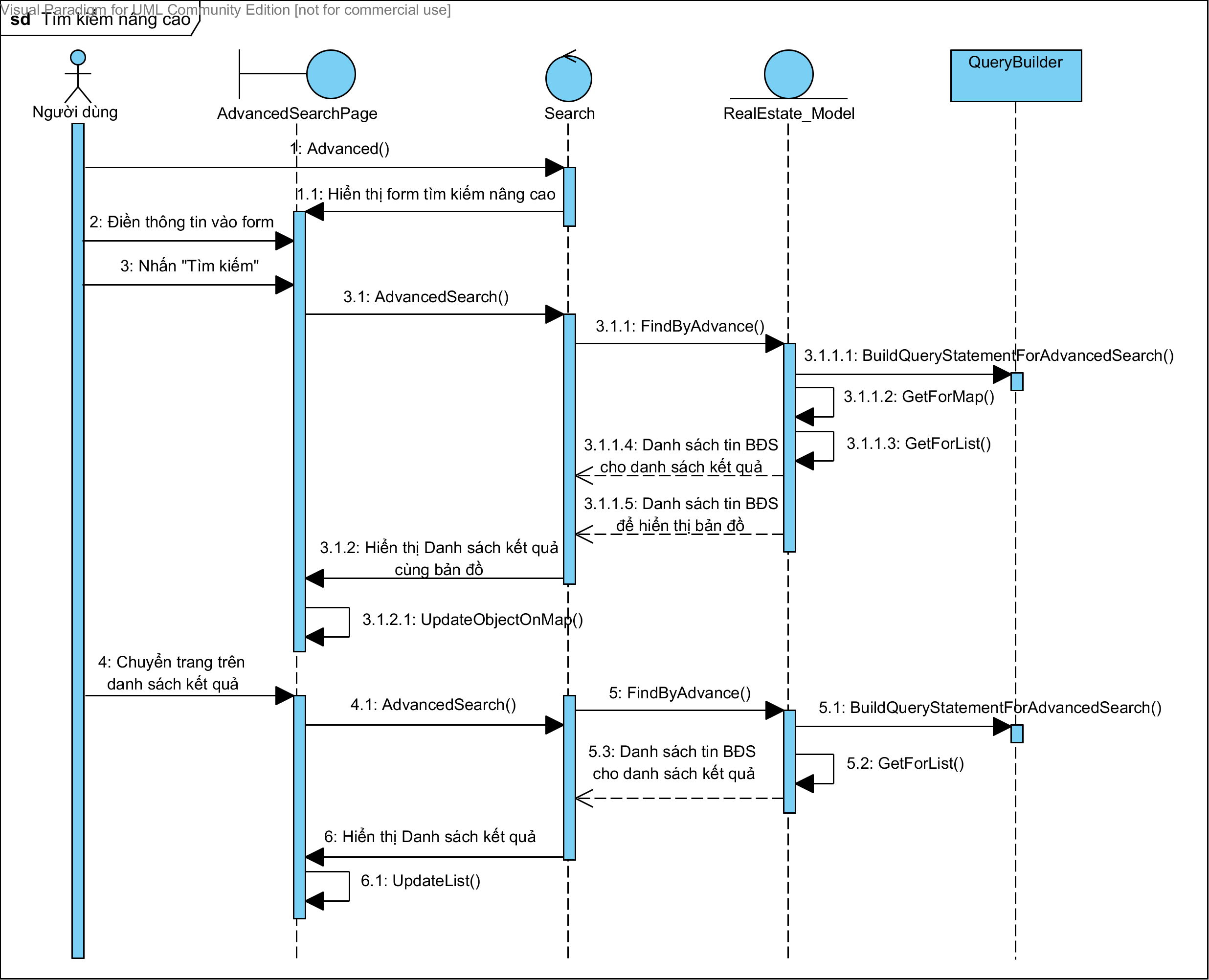
Hình : Sơ đồ trình tự – Tương tác bản đồ

* **Tìm kiếm dữ liệu thuộc tính**

****

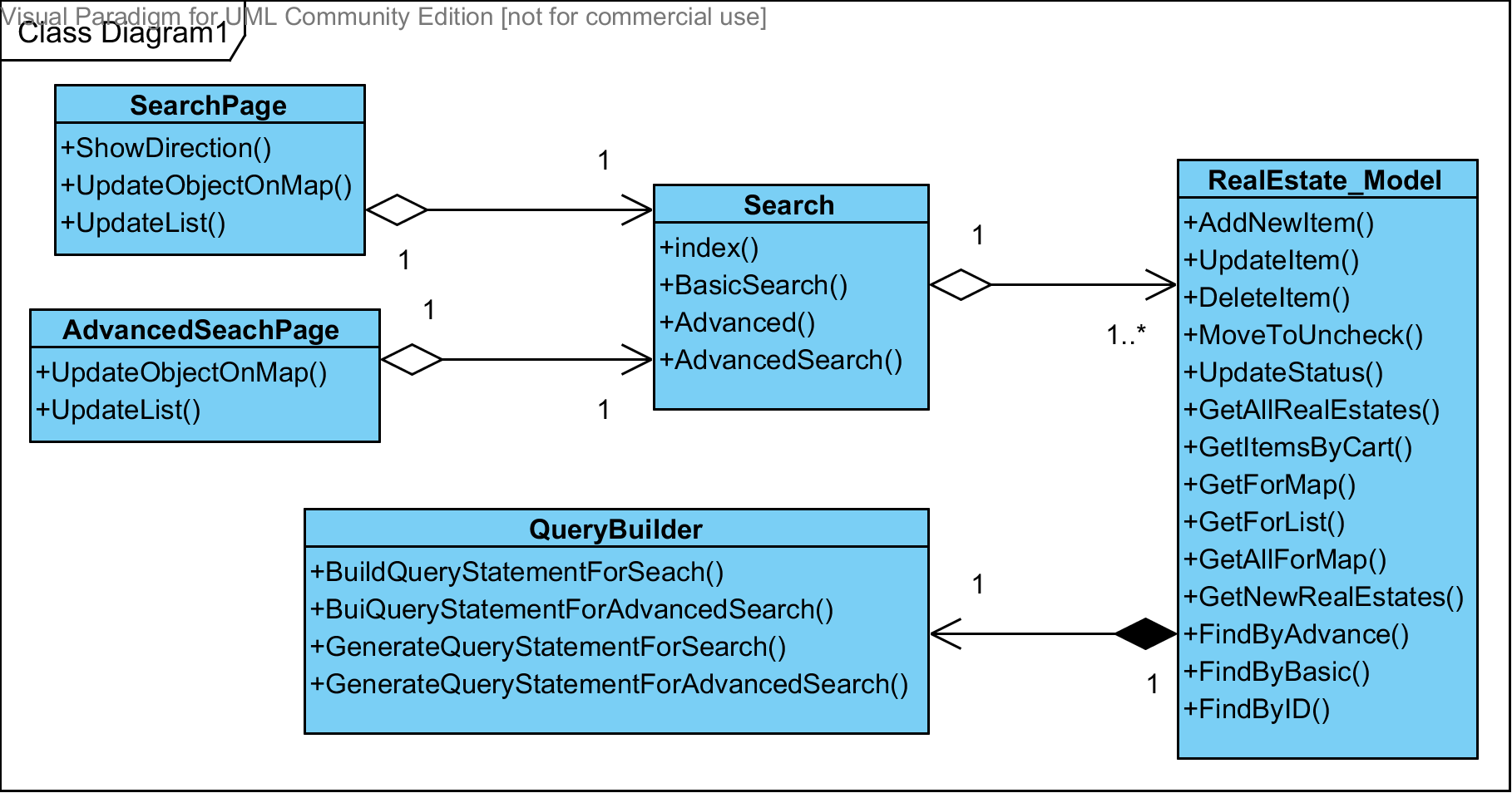
Hình : Sơ đồ trình tự – Tìm kiếm dữ liệu thuộc tính

* **Tìm kiếm không gian**

****

Hình : Sơ đồ trình tự – Tìm kiếm không gian

* + - * 1. **Tìm kiếm – VOPC**

****

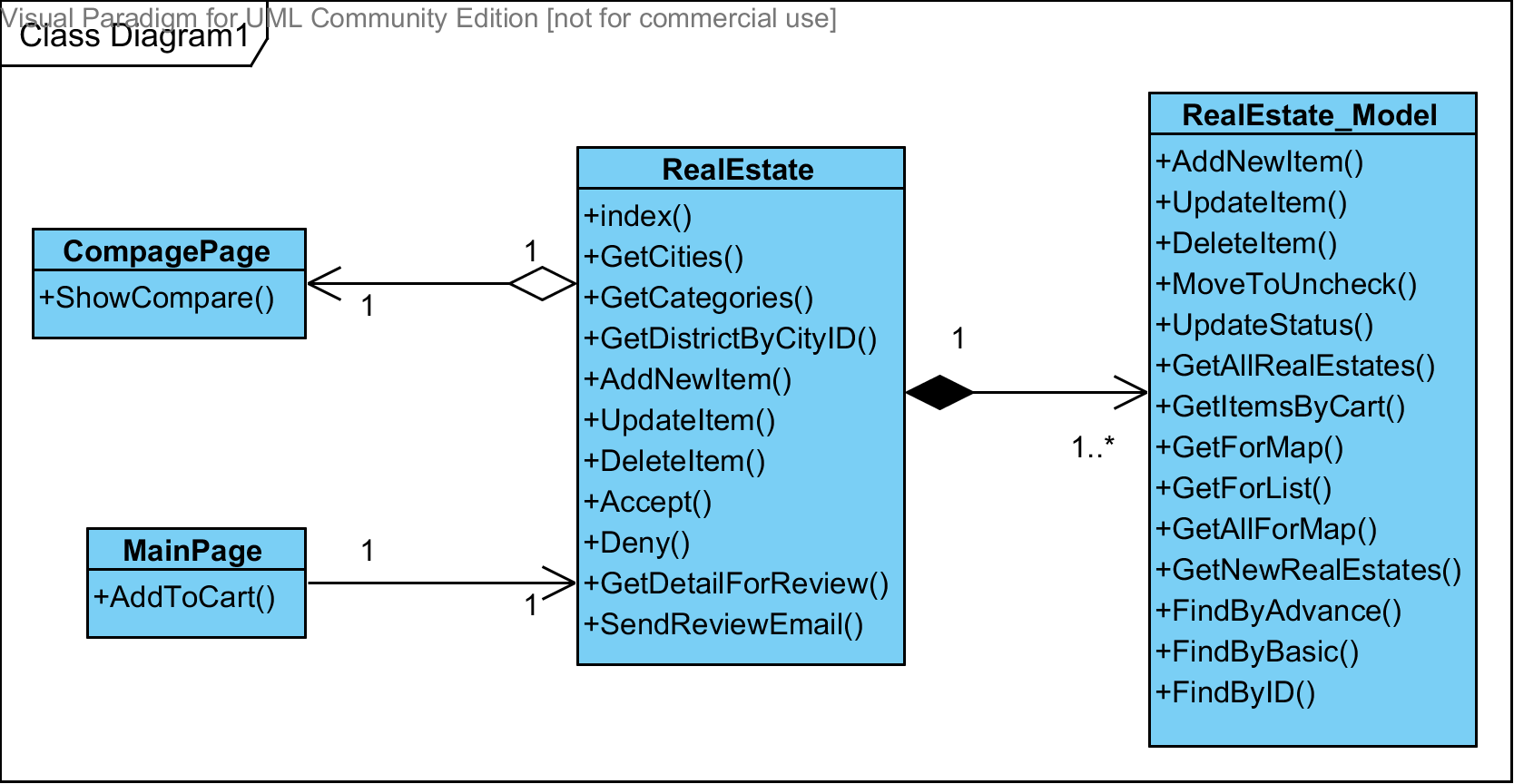
Hình : Sơ đồ lớp – Tìm kiếm

* + - 1. **Usecase So sánh bất động sản**
         1. **So sánh bất động sản – Basic Flow**

****

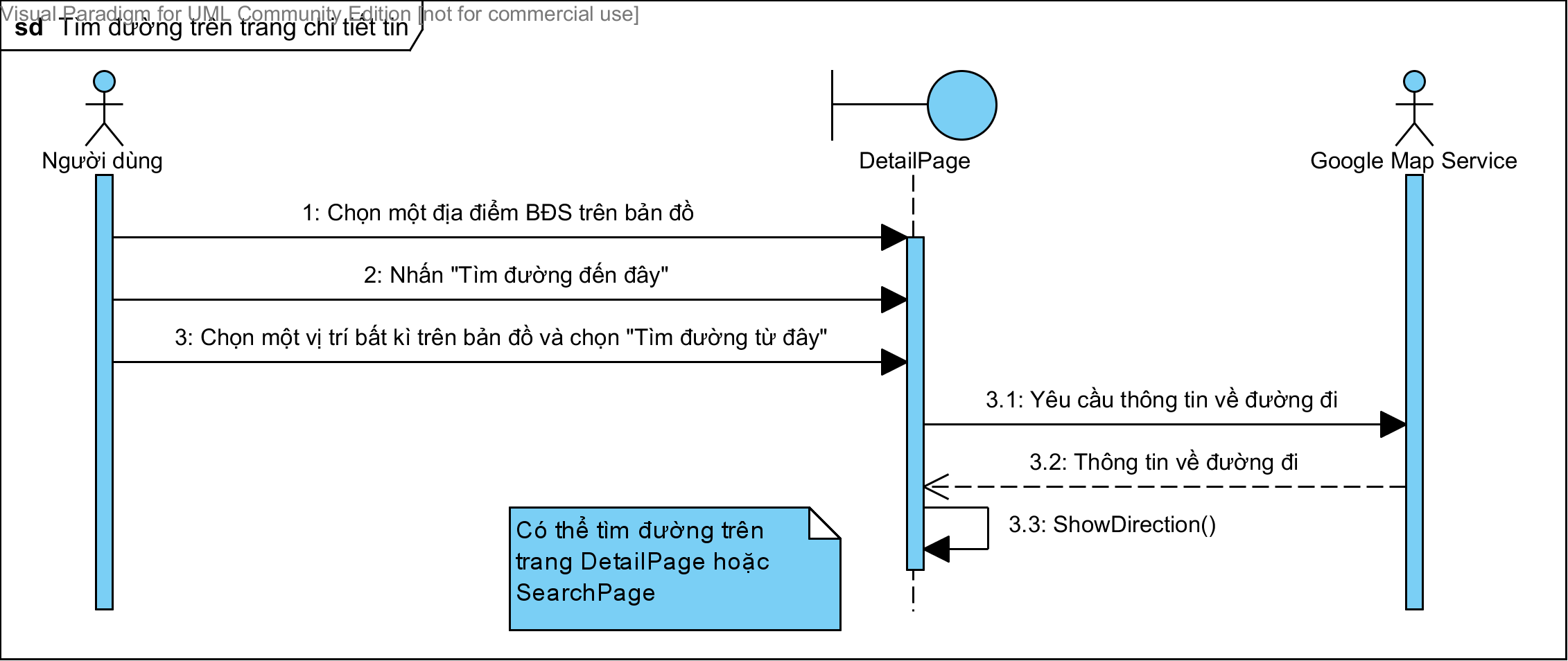
Hình : Sơ đồ trình tự – So sánh bất động sản

* + - * 1. **So sánh bất động sản – VOPC**

****

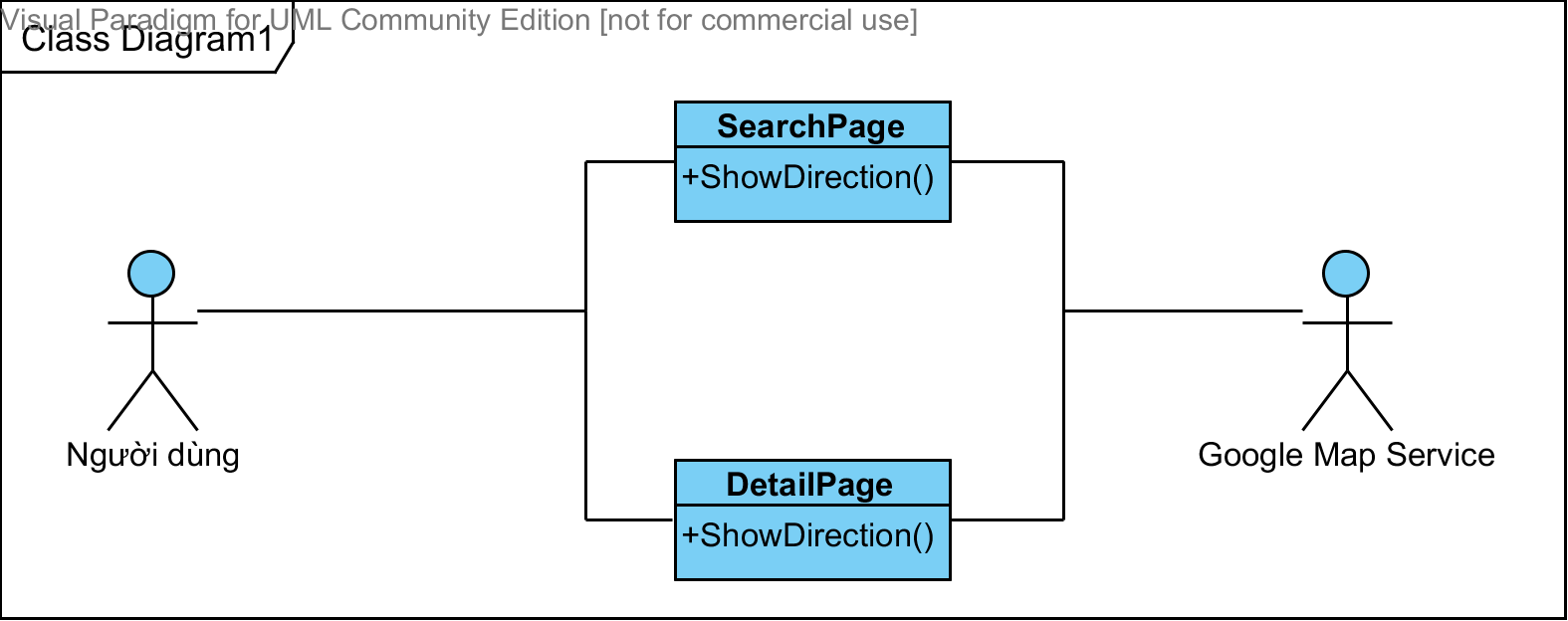
Hình : Sơ đồ lớp – So sánh bất động sản

* + - 1. **Usecase Chỉ dẫn đường đi**
         1. **Chỉ dẫn đường đường đi – Basic Flow**

****

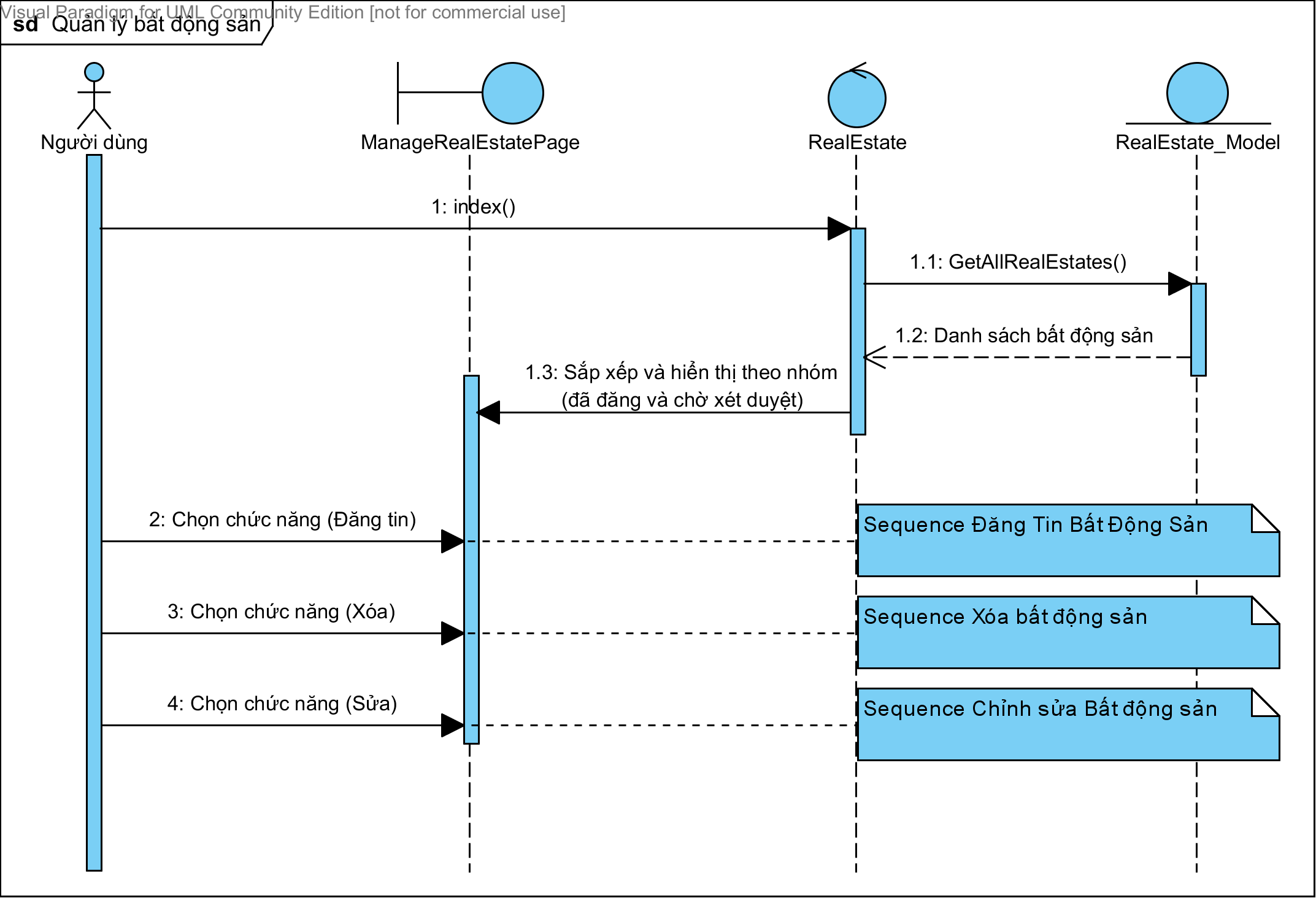
Hình : Sơ đồ trình tự – Chỉ dẫn đường đi

* + - * 1. **Chỉ dẫn đường đi – VOPC**

****

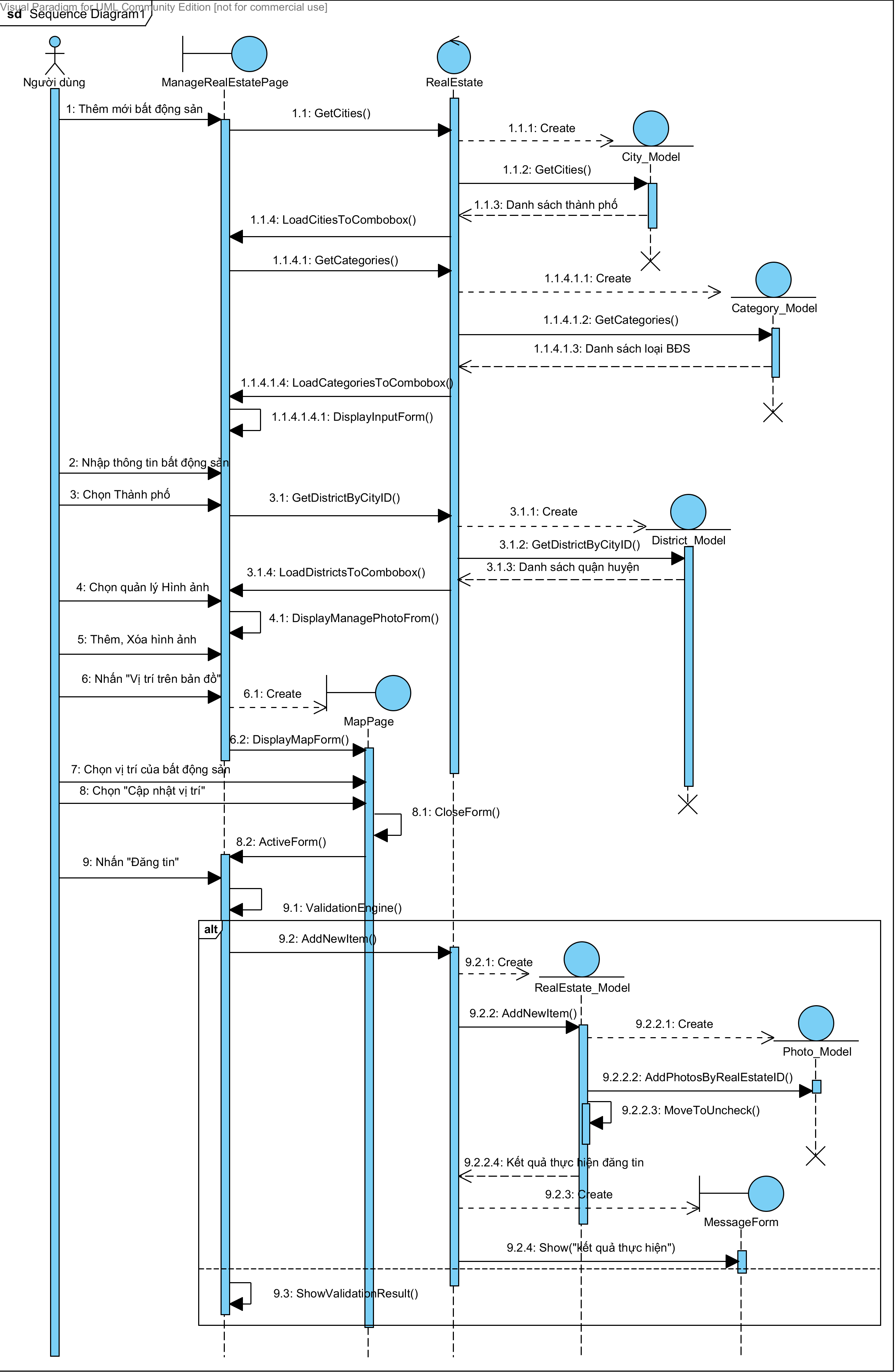
Hình : Sơ đồ lớp – Chỉ dẫn đường đi

* + - 1. **Usecase Quản lý bất động sản**
         1. **Quản lý bất động sản – Basic Flow**

****

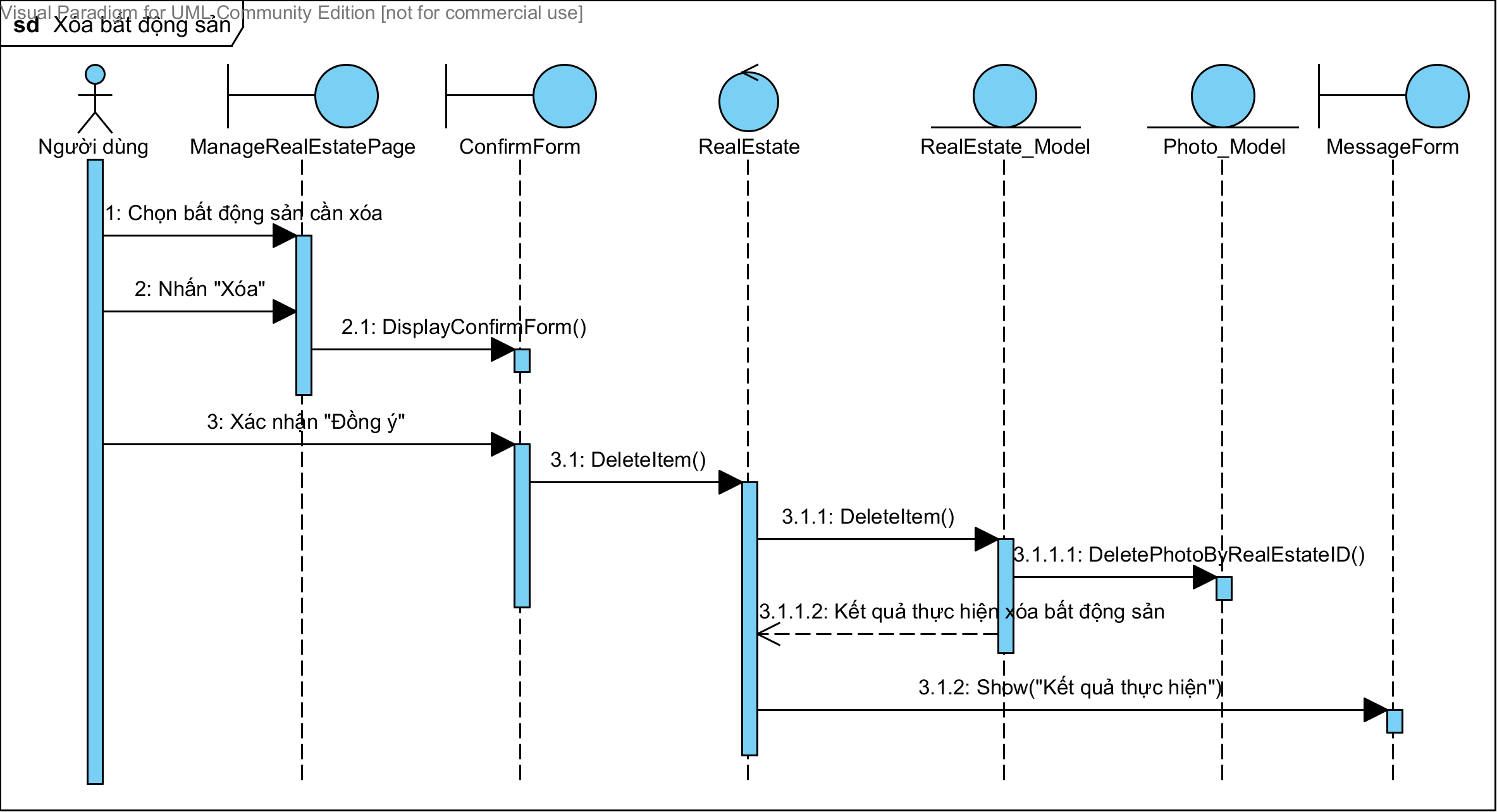
Hình : Sơ đồ trình tự – Quản lý bất động sản

* **Đăng tin**

****

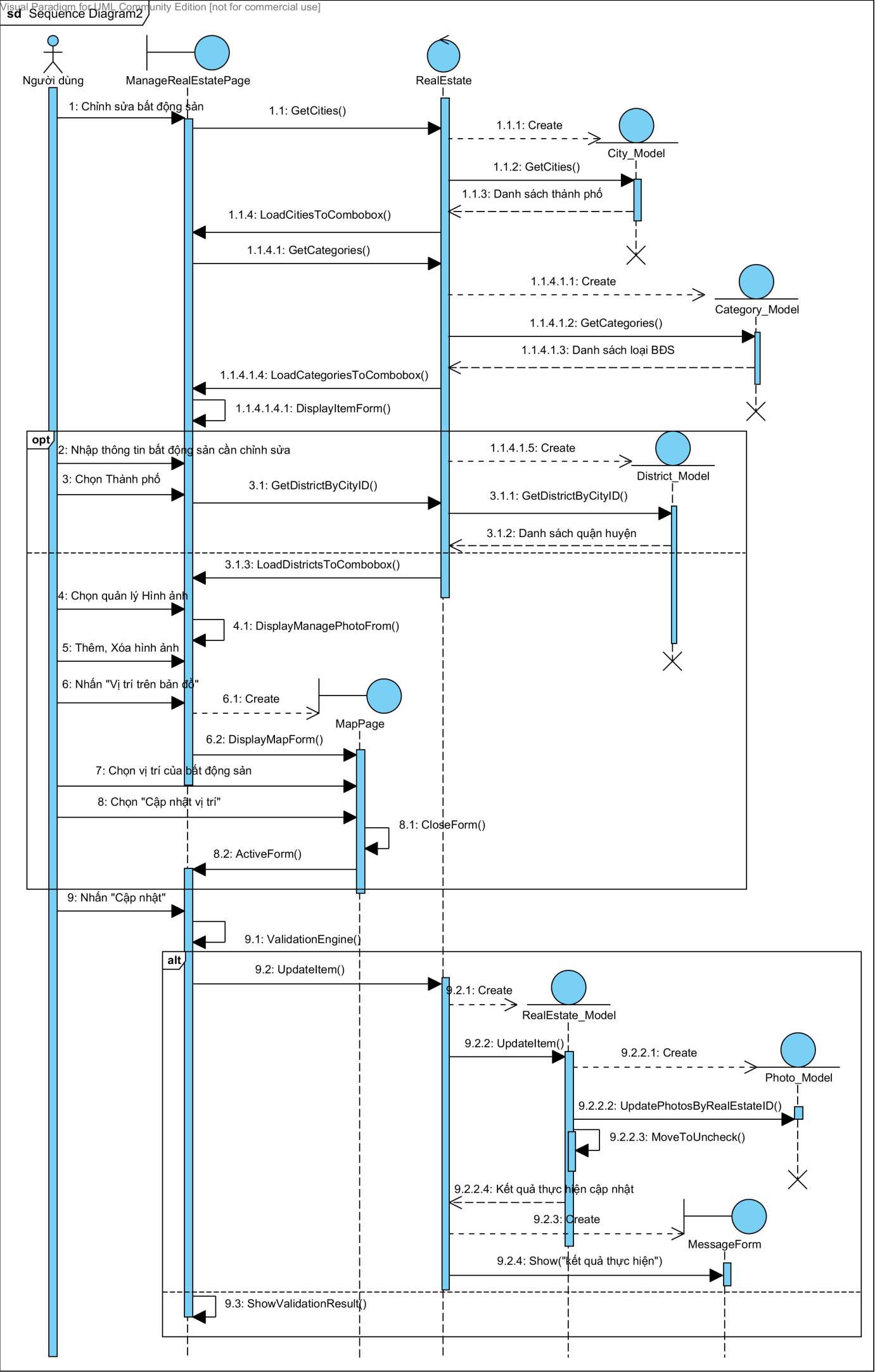
Hình : Sơ đồ trình tự – Đăng tin

* **Xóa tin**

****

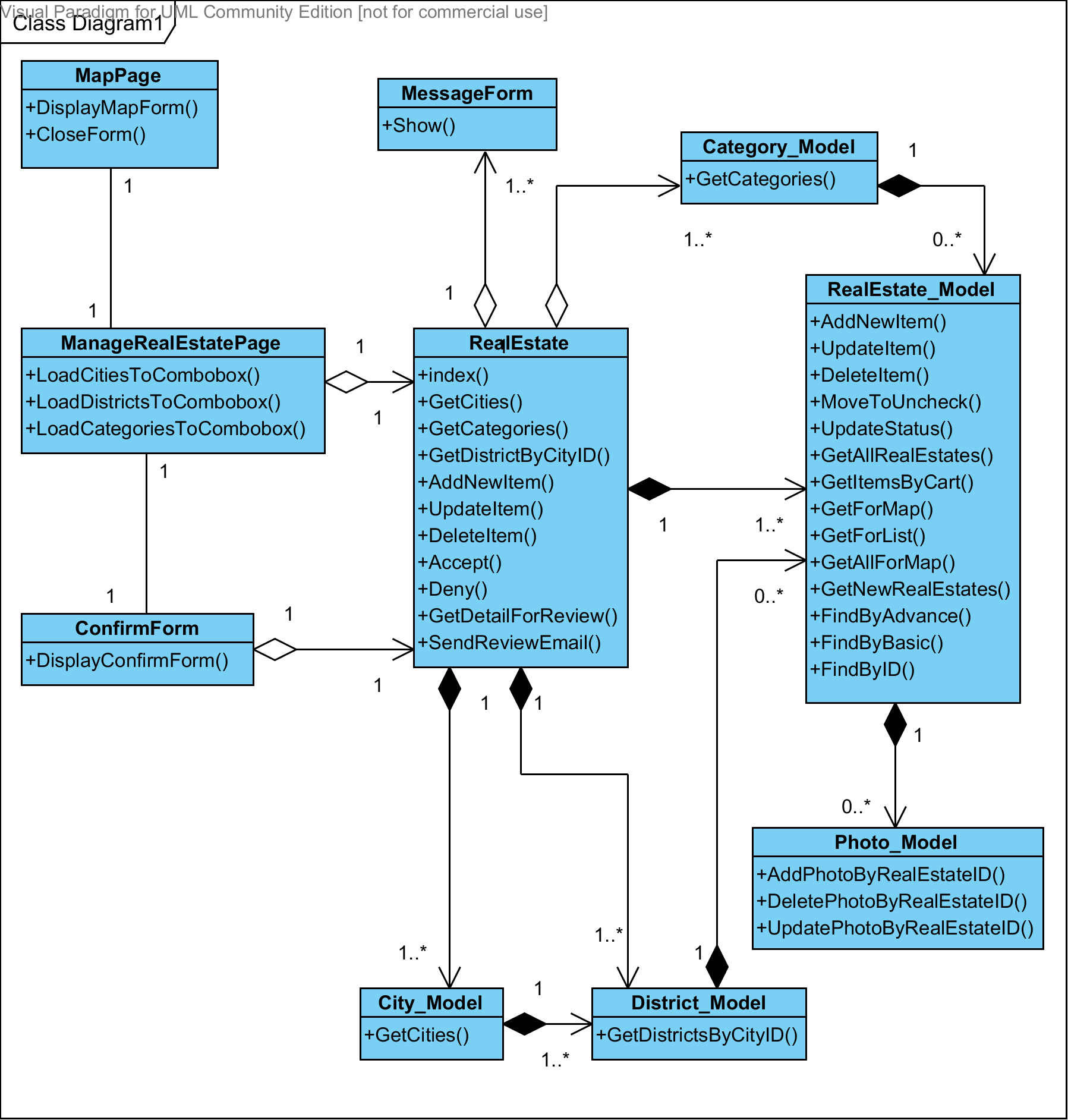
Hình : Sơ đồ trình tự – Xóa tin

* **Chỉnh sửa tin**

****

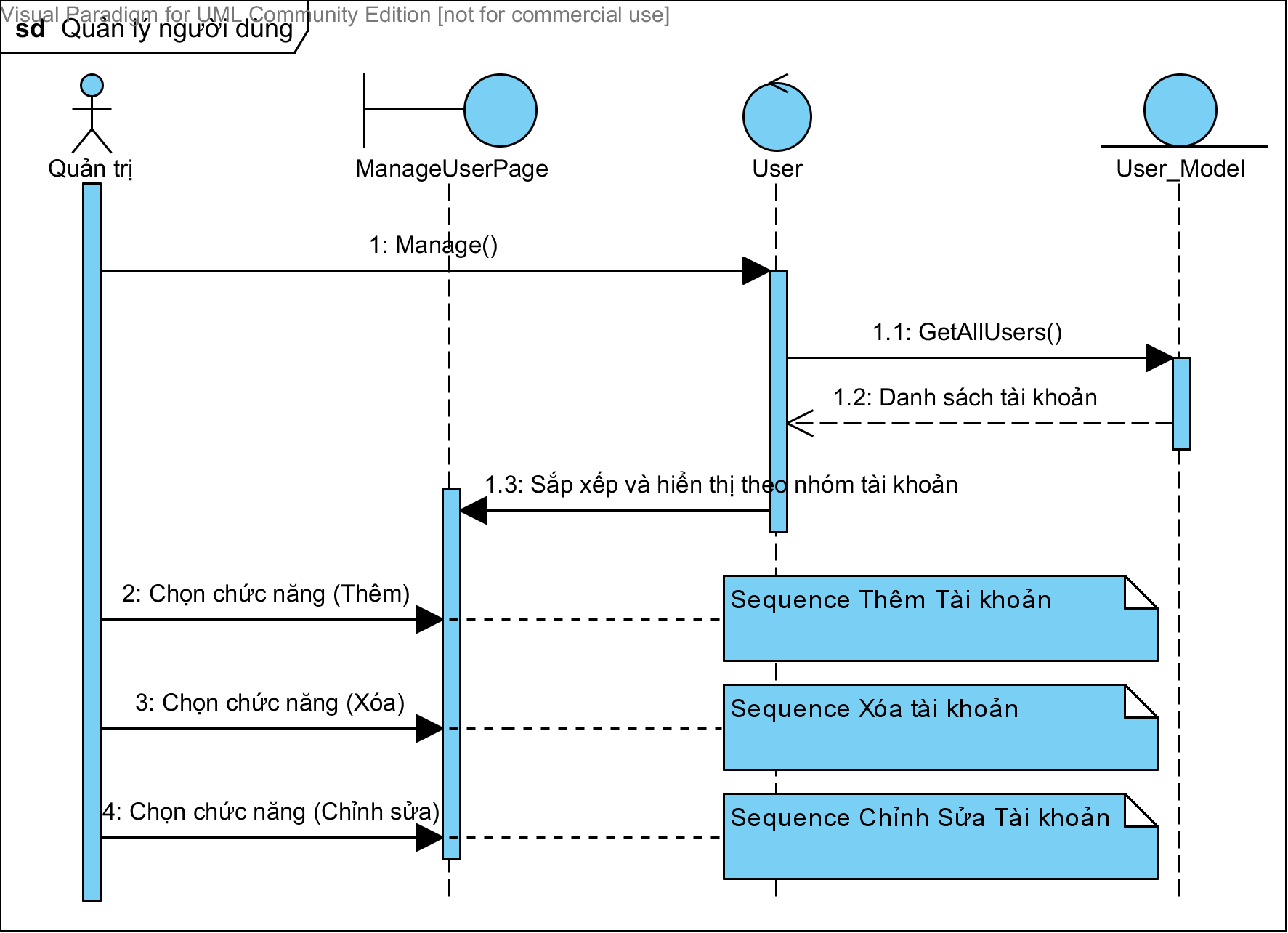
Hình : Sơ đồ trình tự – Chỉnh sửa tin

* + - * 1. **Quản lý bất động sản – VOPC**

****

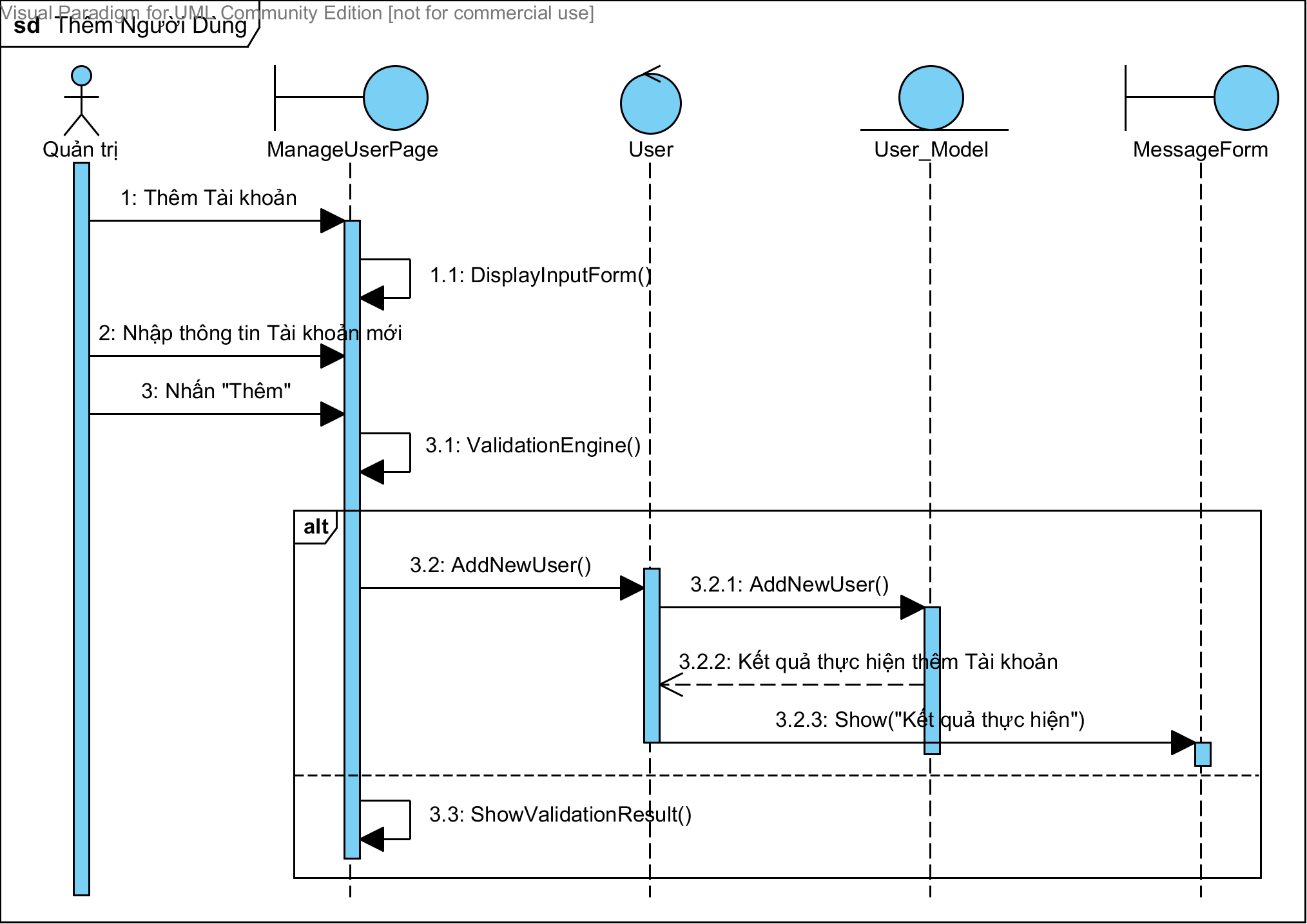
Hình : Sơ đồ lớp – Quản lý bất động sản

* + - 1. **Usecase Quản lý tài khoản**
         1. **Quản lý tài khoản – Basic Flow**

****

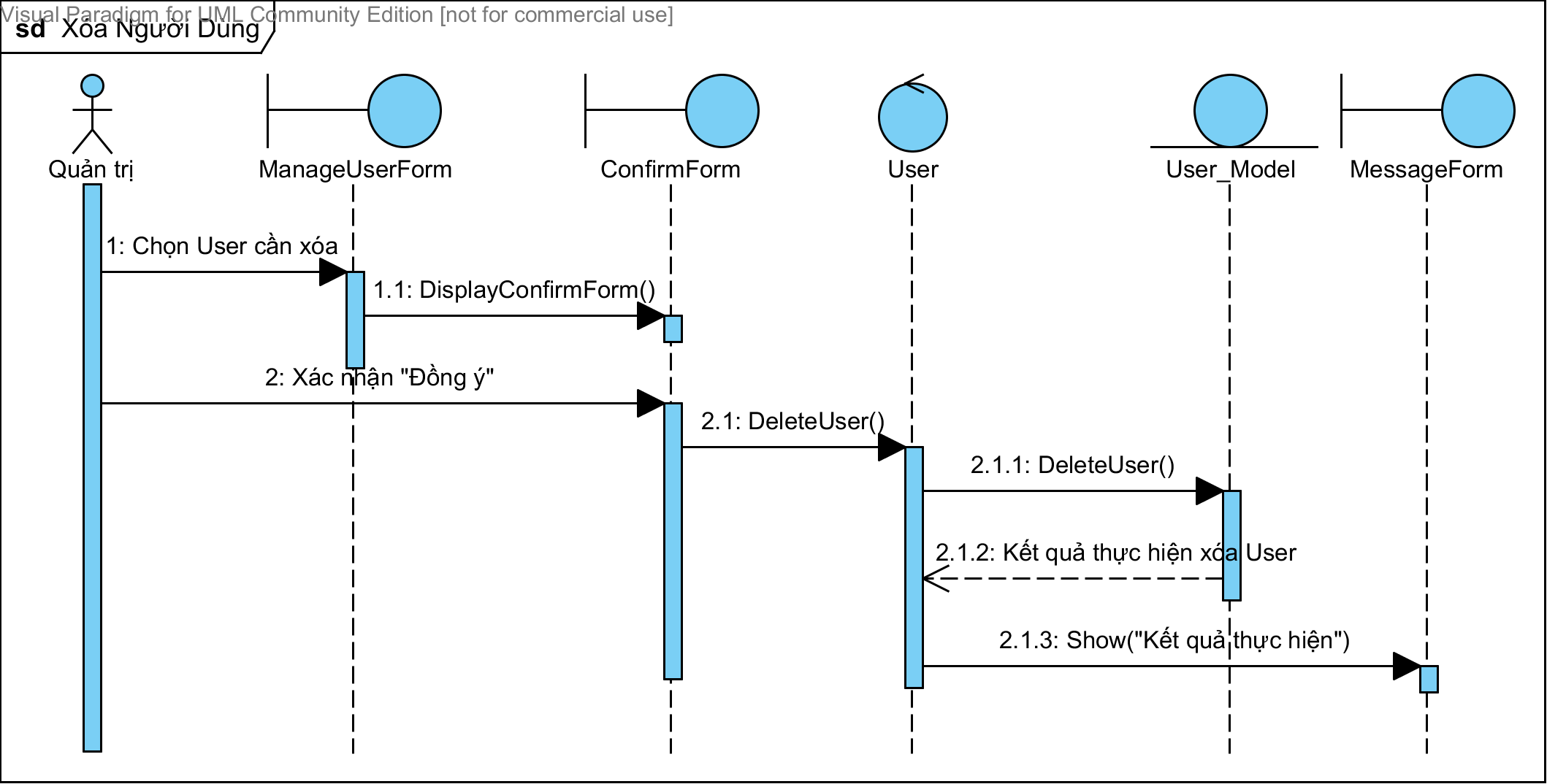
Hình : Sơ đồ trình tự – Quản lý tài khoản

* **Thêm tài khoản**

****

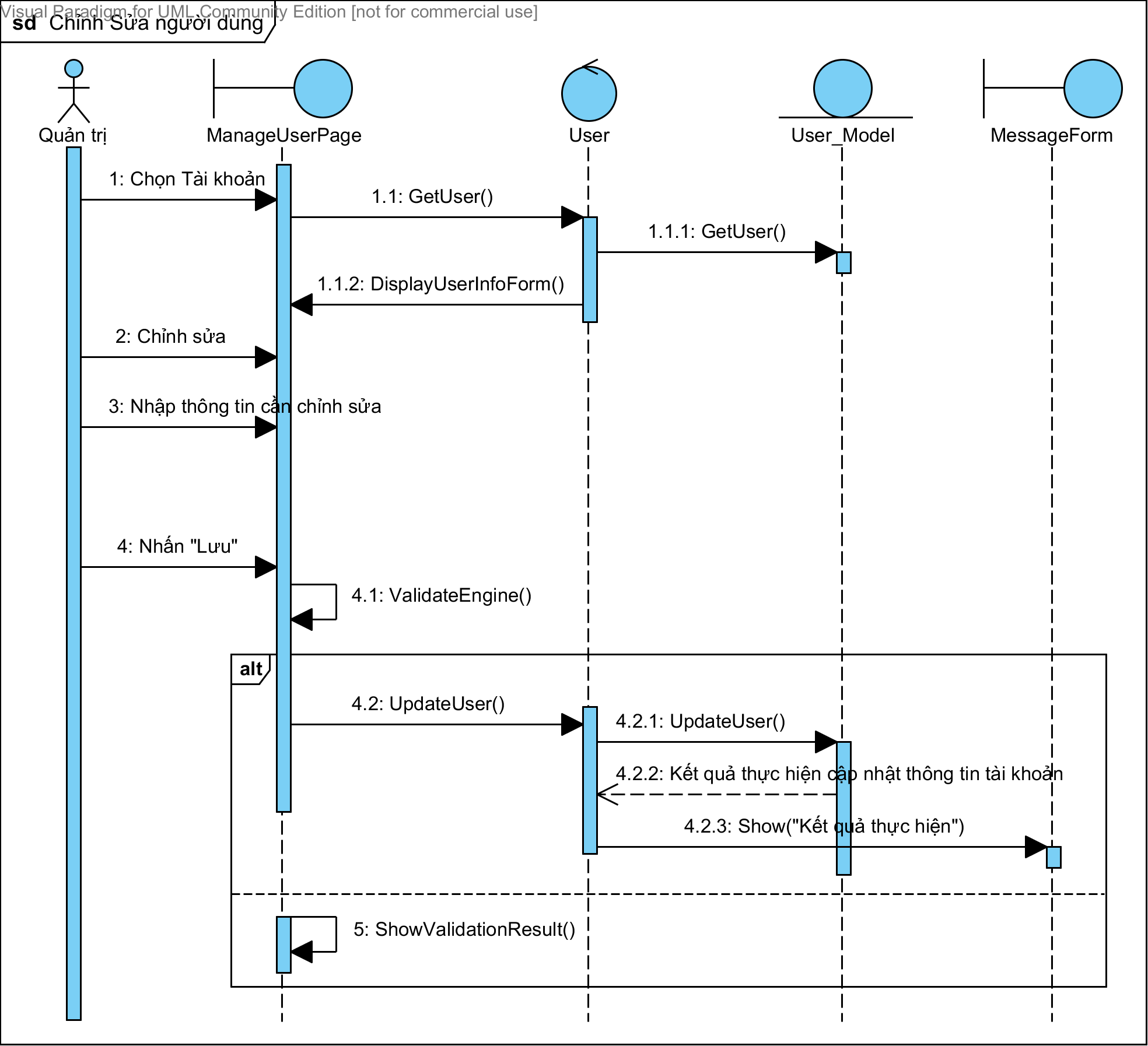
Hình : Sơ đồ trình tự – Thêm tài khoản

* **Xóa tài khoản**

****

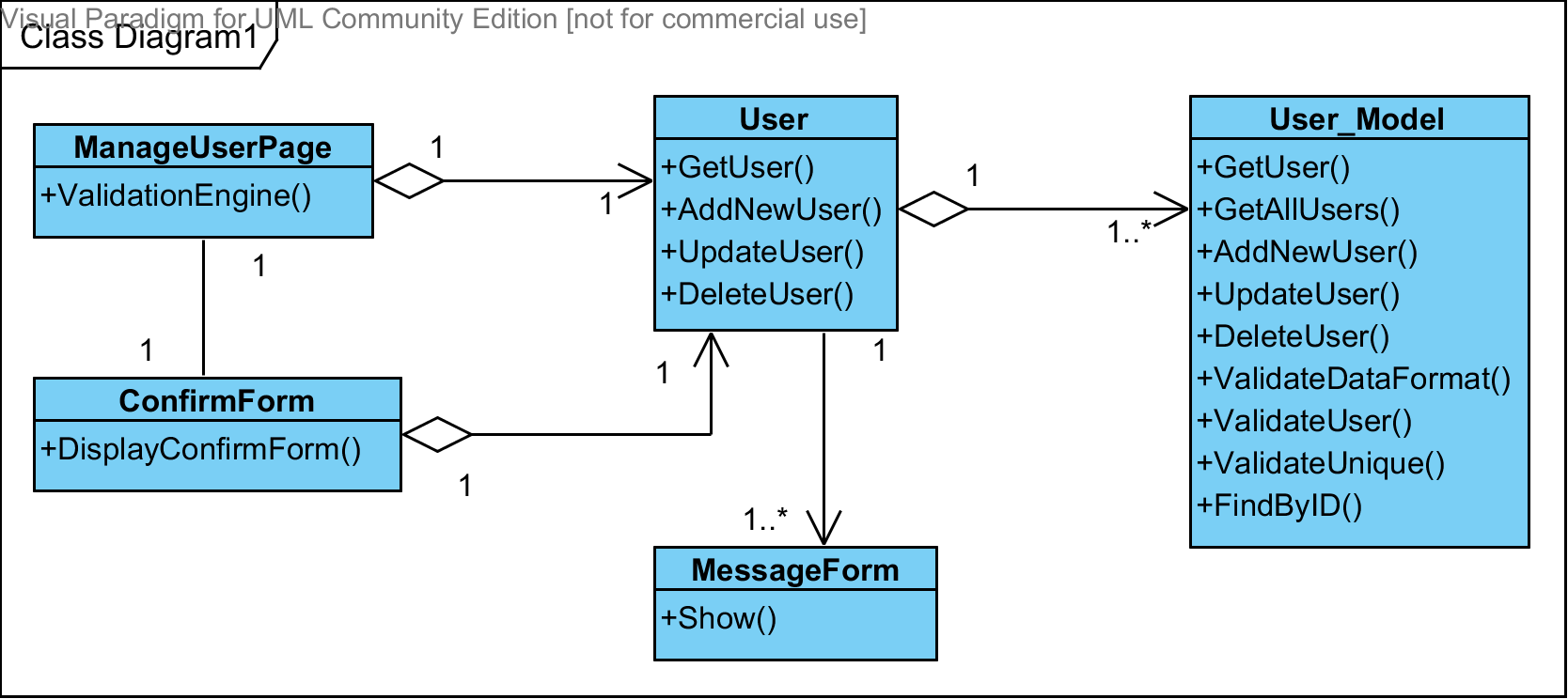
Hình : Sơ đồ trình tự – Xóa tài khoản

* **Chỉnh sửa tài khoản**

****

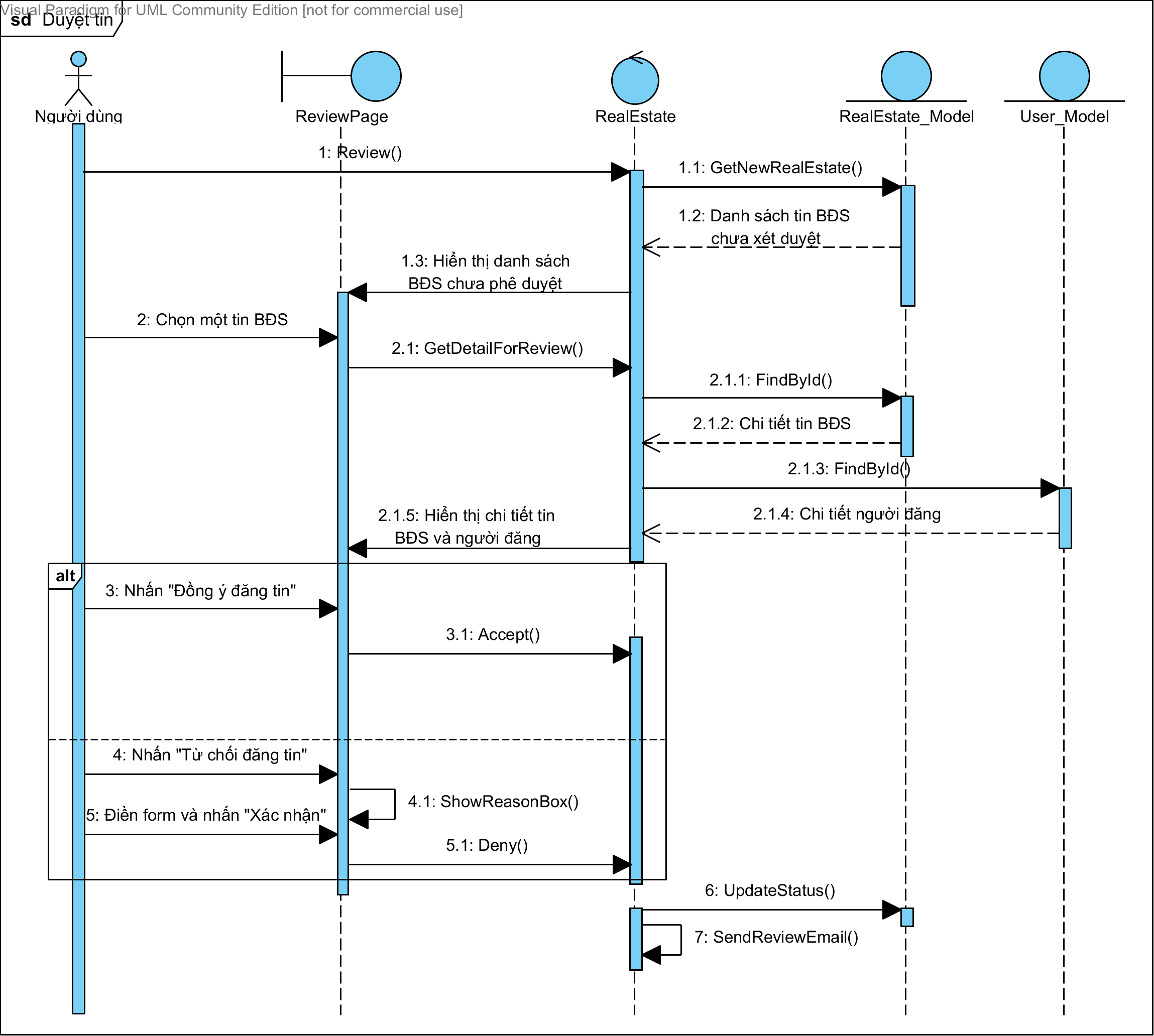
Hình : Sơ đồ trình tự – Chỉnh sửa tài khoản

* + - * 1. **Quản lý tài khoản – VOPC**

****

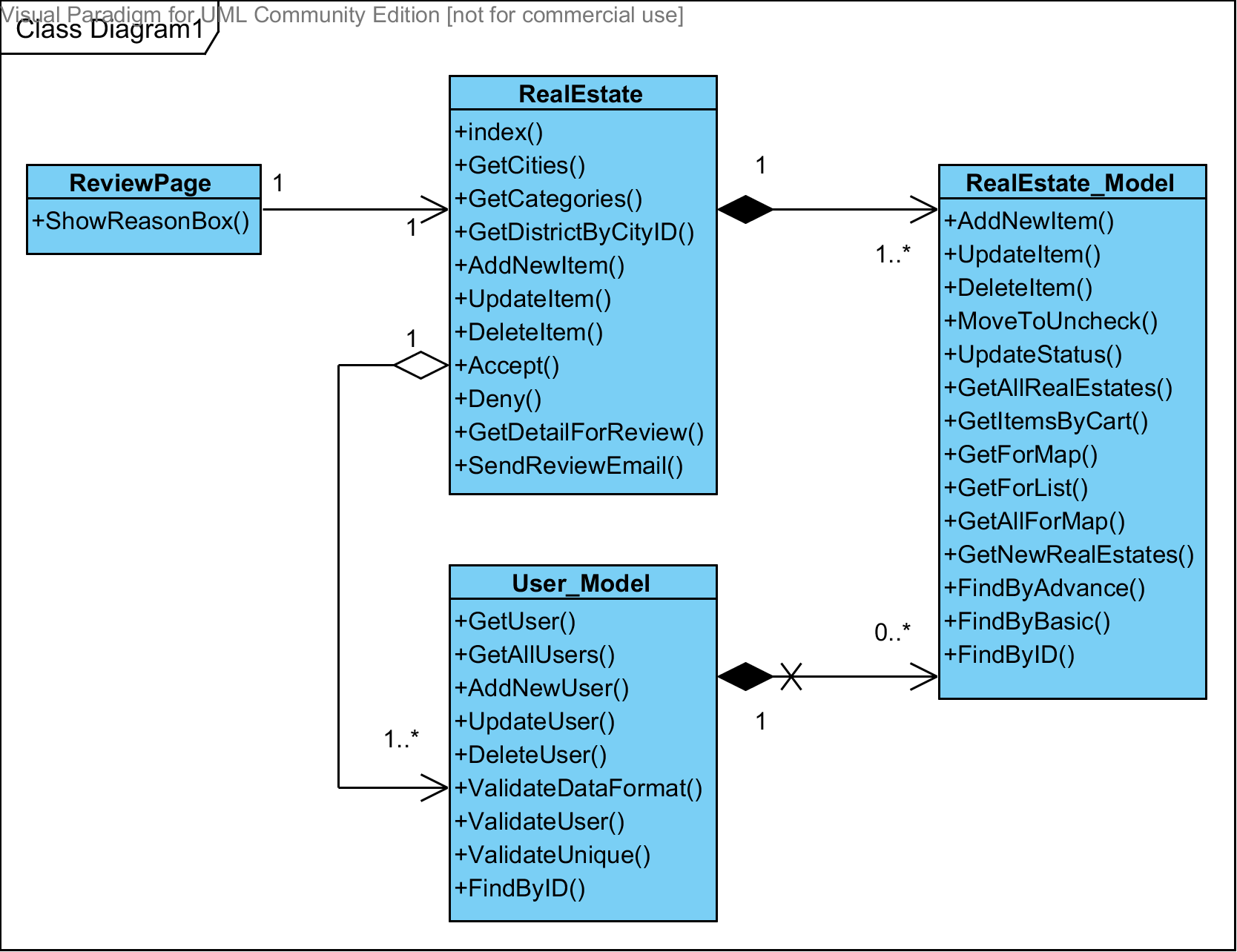
Hình : Sơ đồ lớp – Quản lý tài khoản

* + - 1. **Usecase Duyệt bài**
         1. **Duyệt bài – Basic Flow**

****

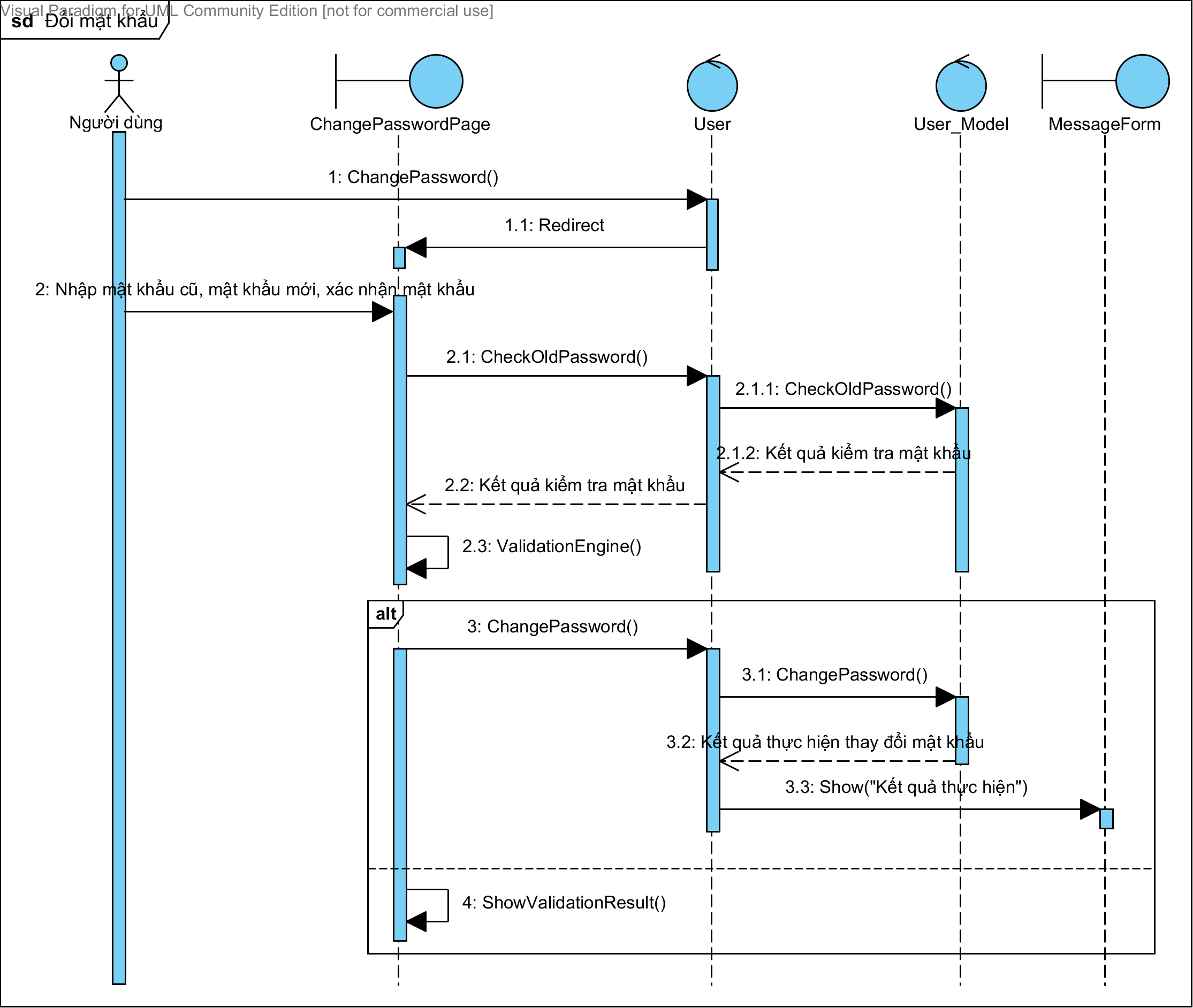
Hình : Sơ đồ trình tự – Duyệt bài

* + - * 1. **Duyệt bài – VOPC**

****

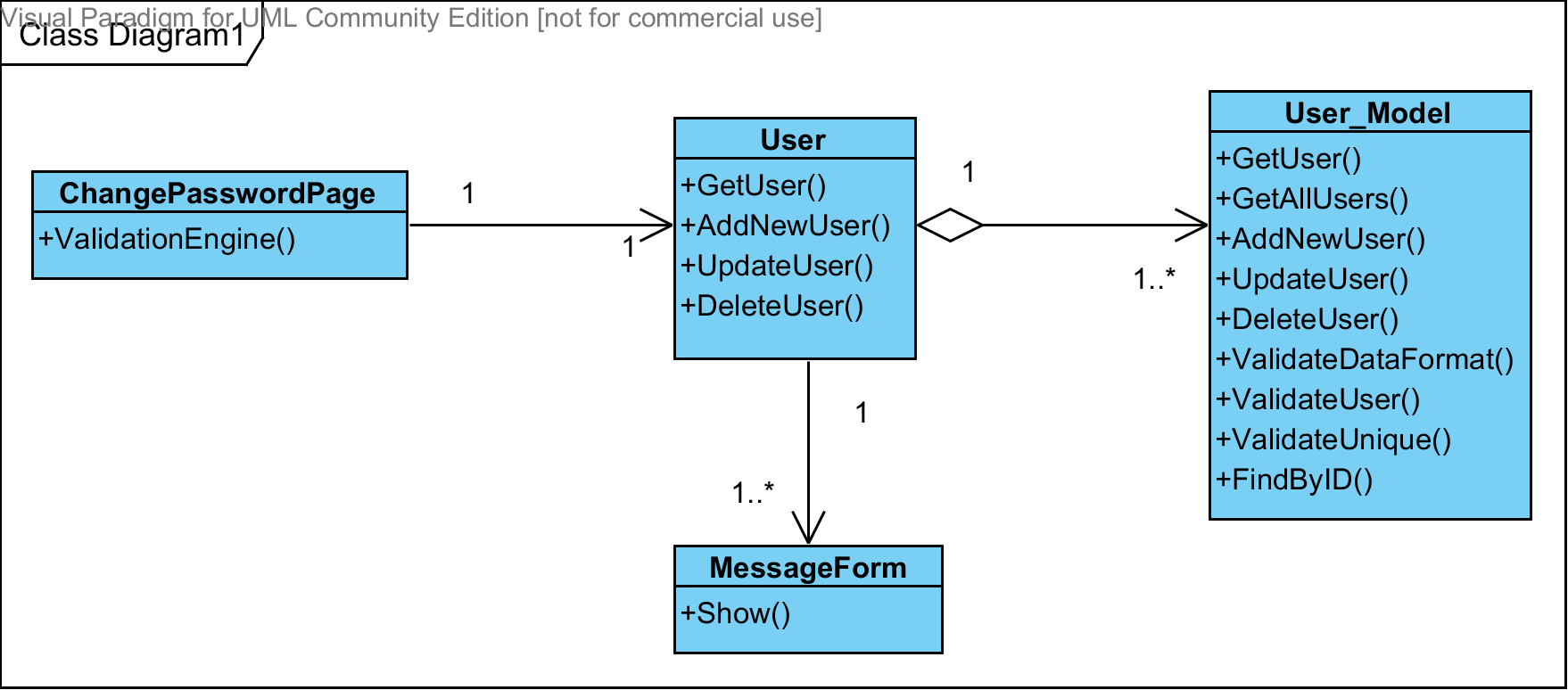
Hình : Sơ đồ lớp – Duyệt bài

* + - 1. **Usecase Thay đổi mật khẩu**
         1. **Thay đổi mật khẩu – Basic Flow**

****

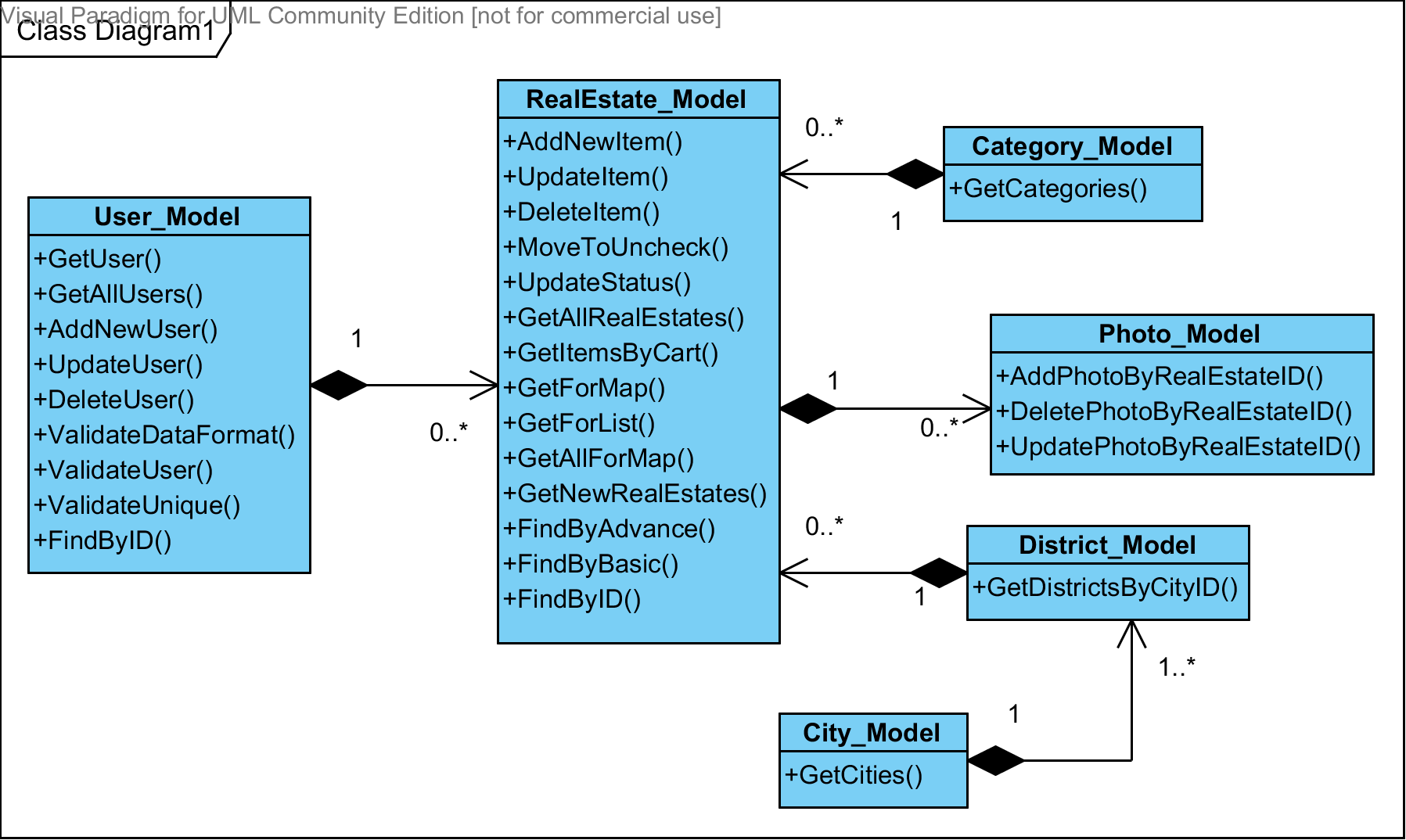
Hình : Sơ đồ trình tự – Thay đổi mật khẩu

* + - * 1. **Thay đổi mật khẩu – VOPC**

****

Hình : Sơ đồ lớp – Thay đổi mật khẩu

# Sơ đồ lớp tổng quát của hệ thống



Hình : Sơ đồ lớp tổng quát của hệ thống

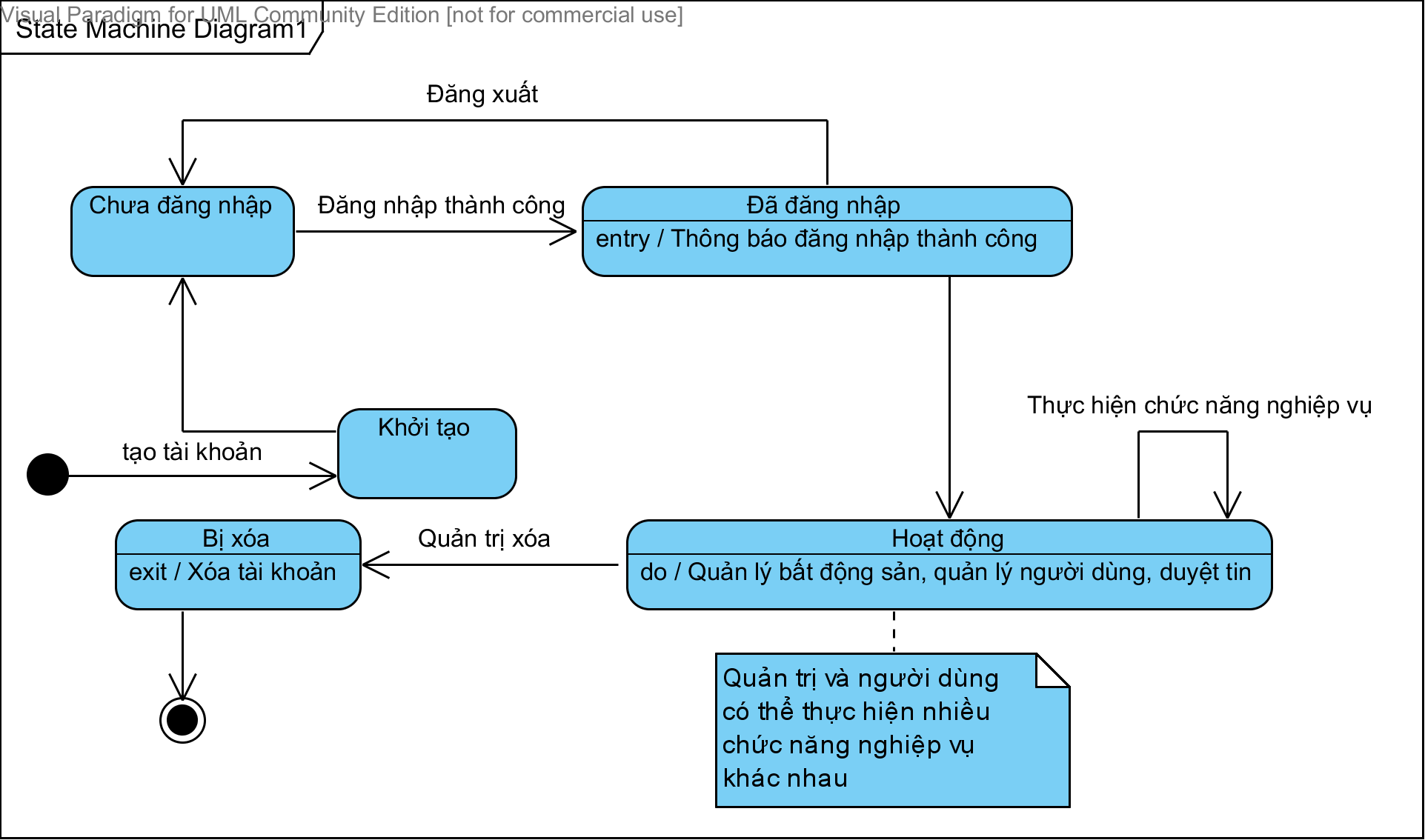
# Danh sách các lớp trong mô hình

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Nhóm** | **Tên Lớp** | **Ý Nghĩa** | **Stereotype** |
| 1 | Model | User\_Model | Thông tin tài khoản | Entity |
| 2 | Model | RealEstate\_Model | Thông tin bất động sản | Entity |
| 3 | Model | Category\_Model | Thông tin loại bất động sản | Entity |
| 4 | Model | City\_Model | Thông tin thành phố/tỉnh | Entity |
| 5 | Model | District\_Model | Thông tin quận/huyện | Entity |
| 6 | Model | Photo\_Model | Thông tin hình ảnh | Entity |
| 7 | Controller | User | Xử lý quản lý tài khoản | Control |
| 8 | Controller | Register | Xử lý đăng ký | Control |
| 9 | Controller | Login | Xử lý đăng nhập | Control |
| 10 | Controller | Search | Xử lý tìm kiếm | Control |
| 11 | Controller | RealEstate | Xử lý quản lý tin bất động sản | Control |
| 12 | View | RegisterPage | Trang đăng ký | Boundary |
| 13 | View | UerNoticePage | Trang thông báo | Boundary |
| 14 | View | LoginPage | Trang đăng nhập | Boundary |
| 15 | View | MessageForm | Form thông báo | Boundary |
| 16 | View | SearchPage | Trang tìm kiếm thuộc tính | Boundary |
| 17 | View | AdvancedSearchPage | Trang tìm kiếm không gian | Boundary |
| 18 | View | MainPage | Trang Chủ | Boundary |
| 19 | View | DetailPage | Trang chi tiết Bất động sản | Boundary |
| 20 | View | ManageRealEstatePage | Trang quản lý bất động sản | Boundary |
| 21 | View | MapPage | Trang hiển thị bản đồ | Boundary |
| 22 | View | ConfirmForm | Form xác nhận | Boundary |
| 23 | View | ManageUserPage | Trang quản lý tài khoản | Boundary |
| 24 | View | ReviewPage | Trang xét duyện tin bất động sản | Boundary |
| 25 | View | ChangePasswordPage | Trang thay đổi mật khẩu | Boundary |
| 26 | View | ComparePage | Trang so sánh | Boundary |
| 27 | Library | QueryBuilder | Xử lý truy vấn | ClassUtility |

Bảng : Danh sách các lớp trong hệ thống

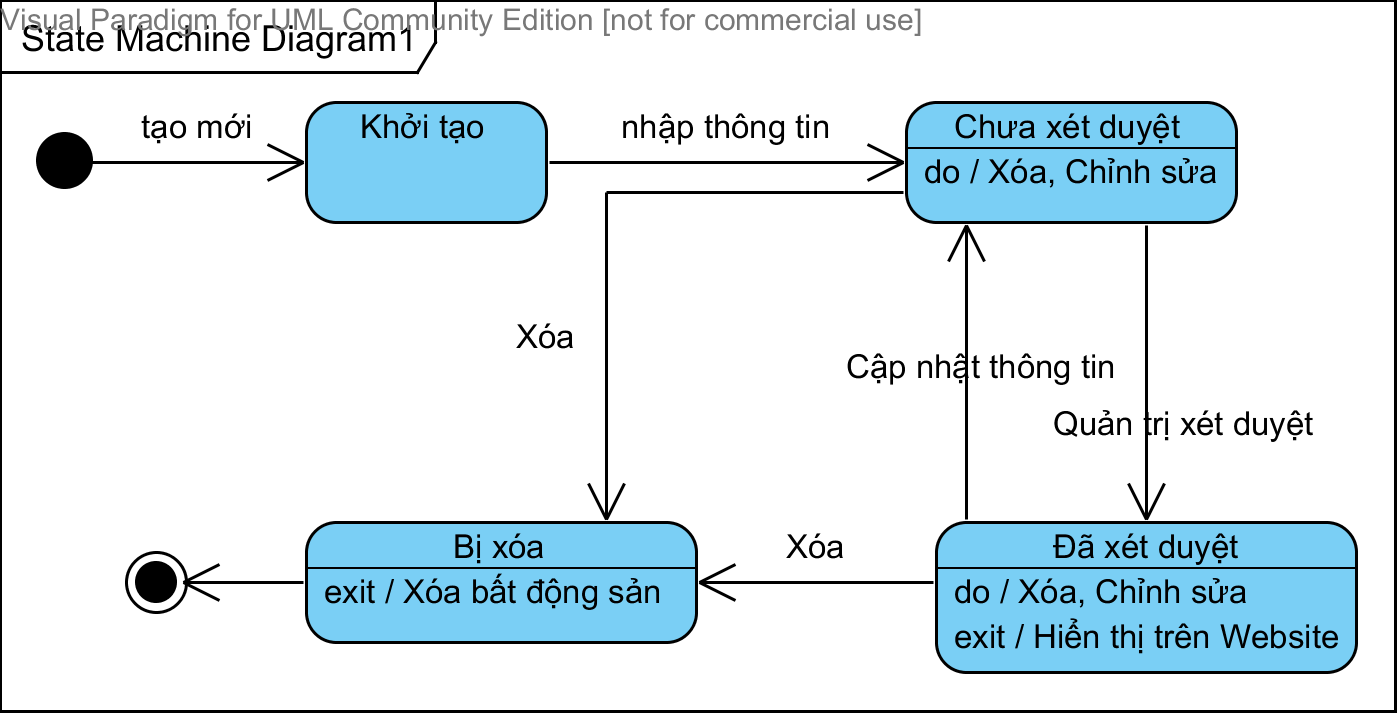
# Biểu đồ trạng thái (State Diagram)

* + 1. **Entity “User”**

****

Hình : Sơ đồ trạng thái Entity “User”

* + 1. **Entity “RealEstate”**

****

Hình : Sơ đồ trạng thái Entity “RealEstate”

# Thiết kế CSDL

# Mô hình dữ liệu mức vật lý



Hình : Mô hình dữ liệu mức vật lý

# Mô tả các bảng

* + - 1. **Bảng bất động sản (RealEstate)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường (Name)** | **Kiểu dữ liệu (Type)** | **Độ rộng (Width)** | **Mô tả** |
| **1** | RealEstateID | INT4 |  | Mã bất động sản |
| **2** | Title | VARCHAR | 200 | Tiêu đề |
| **3** | Address | VARCHAR | 500 | Địa chỉ |
| **4** | Price | DECIMAL |  | Giá |
| **5** | Currency | VARCHAR | 10 | Đơn vị tiền tệ |
| **6** | Unit | VARCHAR | 10 | Đơn vị tính |
| **7** | Area | DECIMAL |  | Diện tích |
| **8** | Size | VARCHAR | 50 | Kích thước |
| **9** | Direction | VARCHAR | 50 | Phương hướng |
| **10** | Alley | DECIMAL |  | Độ rộng của đường hẻm (nếu bất động sản là nhà, căn hộ) |
| **11** | LegalStatus | VARCHAR | 100 | Tình trạng pháp lý |
| **12** | Transaction | VARCHAR | 20 | Loại giao dịch |
| **13** | Description | VARCHAR | 1000 | Mô tả chi tiết |
| **14** | Date | DATE |  | Ngày đăng |
| **15** | DateModified | DATE |  | Ngày chỉnh sửa lần cuối |
| **16** | Reason | VARCHAR | 500 | Lý do từ chối đăng (trong trường hợp không được đăng). |
| **17** | Status | VARCHAR | 10 | Tình trạng của bất động sản đã được xét duyệt đăng. |
| **18** | CategoryID | INT4 |  | Mã loại bất động sản |
| **19** | DistrictID | INT4 |  | Mã quận huyện trực thuộc |
| **20** | UserID | INT4 |  | Mã người dùng sở hữu bất động sản |
| **21** | AdminID | INT4 |  | Mã quản trị xét duyệt bất động sản |
| **22** | Geom | POINT  (Geometry) |  | Đối tượng không gian dạng điểm (**point**) |
| **23** | ContactName | VARCHAR | 100 | Người liên hệ |
| **24** | ContactTel | VARCHAR | 100 | Điện thoại liên hệ |
| **25** | ContactAdd | VARCHAR | 500 | Địa chỉ liên hệ |
| **26** | Remark | VARCHAR | 500 | Ghi chú |

Bảng : Danh sách thuộc tính “RealEstate”

* + - 1. **Bảng Người dùng (User)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường (Name)** | **Kiểu dữ liệu (Type)** | **Độ rộng (Width)** | **Mô tả** |
| **1** | UserID | INT4 |  | Mã người dùng |
| **2** | Username | VARCHAR | 50 | Tên đăng nhập |
| **3** | Password | VARCHAR | 50 | Mật khẩu |
| **4** | Email | VARCHAR | 50 | Địa chỉ email |
| **5** | Name | VARCHAR | 100 |  |
| **6** | Gender | VARCHAR | 10 | Giới tính |
| **7** | Brithday | DATE |  | Ngày sinh |
| **8** | Tel | VARCHAR | 15 | Điện thoại |
| **9** | IDNumber | VARCHAR | 10 | Số CMND |
| **10** | Address | VARCHAR | 500 | Địa chỉ |
| **11** | Duty | VARCHAR | 10 | Quyền hạn |

Bảng : Danh sách thuộc tính “User”

* + - 1. **Bảng Quận Huyện (District)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường (Name)** | **Kiểu dữ liệu (Type)** | **Độ rộng (Width)** | **Mô tả** |
| **1** | DistrictID | INT4 |  | Mã quận huyện |
| **2** | Name | VARCHAR | 100 | Tên quận huyện |
| **3** | CityID | INT4 |  | Mã thành phố trực thuộc |

Bảng : Danh sách thuộc tính “District”

* + - 1. **Bảng Thành phố (City)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường (Name)** | **Kiểu dữ liệu (Type)** | **Độ rộng (Width)** | **Mô tả** |
| **1** | CityID | INT4 |  | Mã thành phố |
| **2** | Name | VARCHAR | 100 | Tên thành phố |

Bảng : Danh sách thuộc tính “City”

* + - 1. **Bảng Loại bất động sản (Category)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường (Name)** | **Kiểu dữ liệu (Type)** | **Độ rộng (Width)** | **Mô tả** |
| **1** | CategoryID | INT4 |  | Mã loại bất động sản |
| **2** | Name | VARCHAR | 100 | Tên loại bất động sản |

Bảng : Danh sách thuộc tính “Category”

# Bảng Hình ảnh (Photo)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường (Name)** | **Kiểu dữ liệu (Type)** | **Độ rộng (Width)** | **Mô tả** |
| **1** | PhotoID | INT4 |  | Mã hình ảnh |
| **2** | Url | VARCHAR | 500 | Url của hình ảnh |
| **3** | RealEstateID | INT4 |  | Mã bất động sản liên quan |

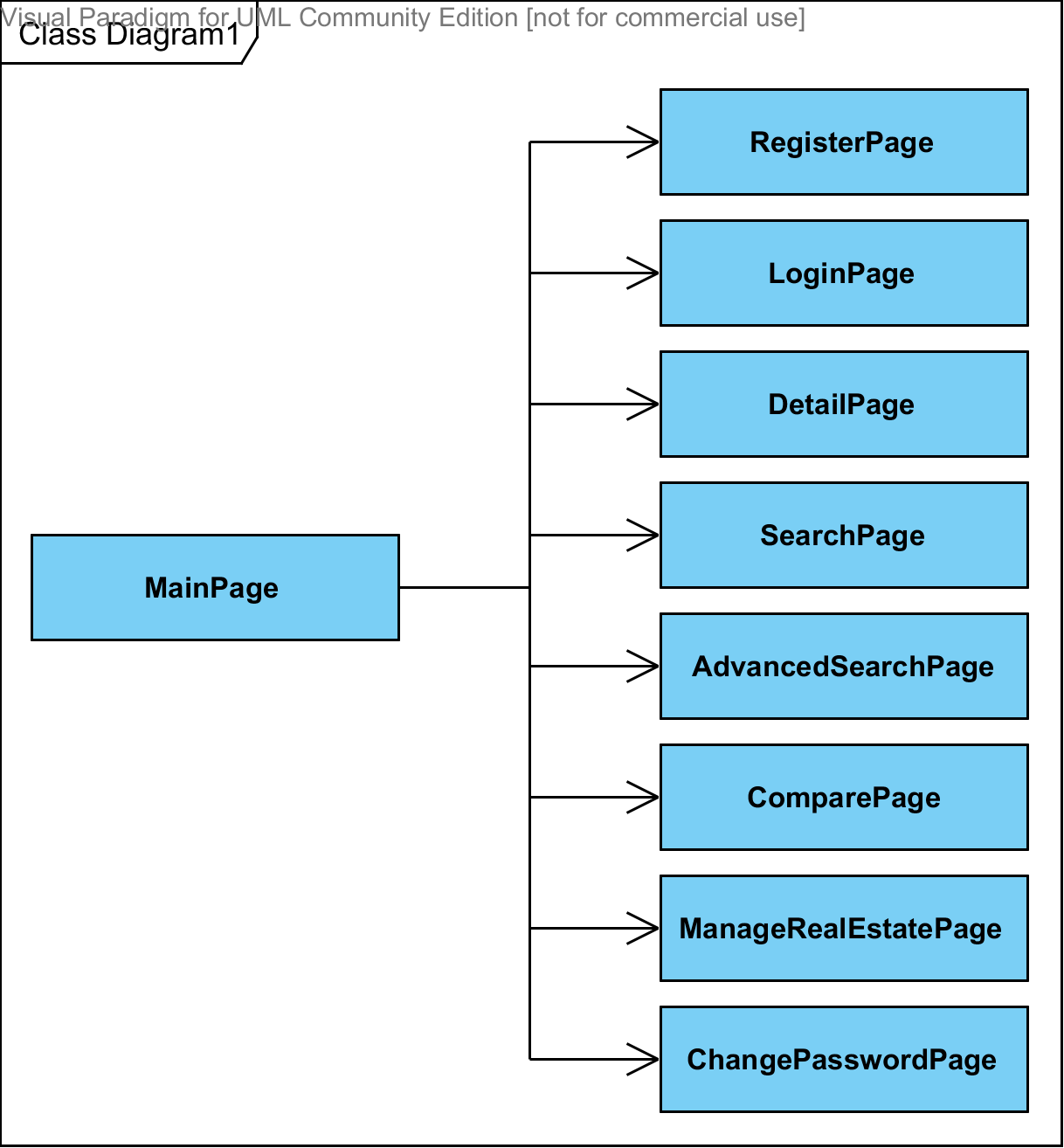
# Các lớp dữ liệu nền trong hệ thống

* Hệ thống sử dụng dữ liệu nền Thành phố Hồ Chí Minh, các lớp dữ liệu nền sau:
  + Lớp đường giao thông
  + Lớp hành chính xã
  + Lớp hành chính huyện
  + Lớp hành chính tỉnh
  + Lớp địa vật

# Thiết kế giao diện chương trình

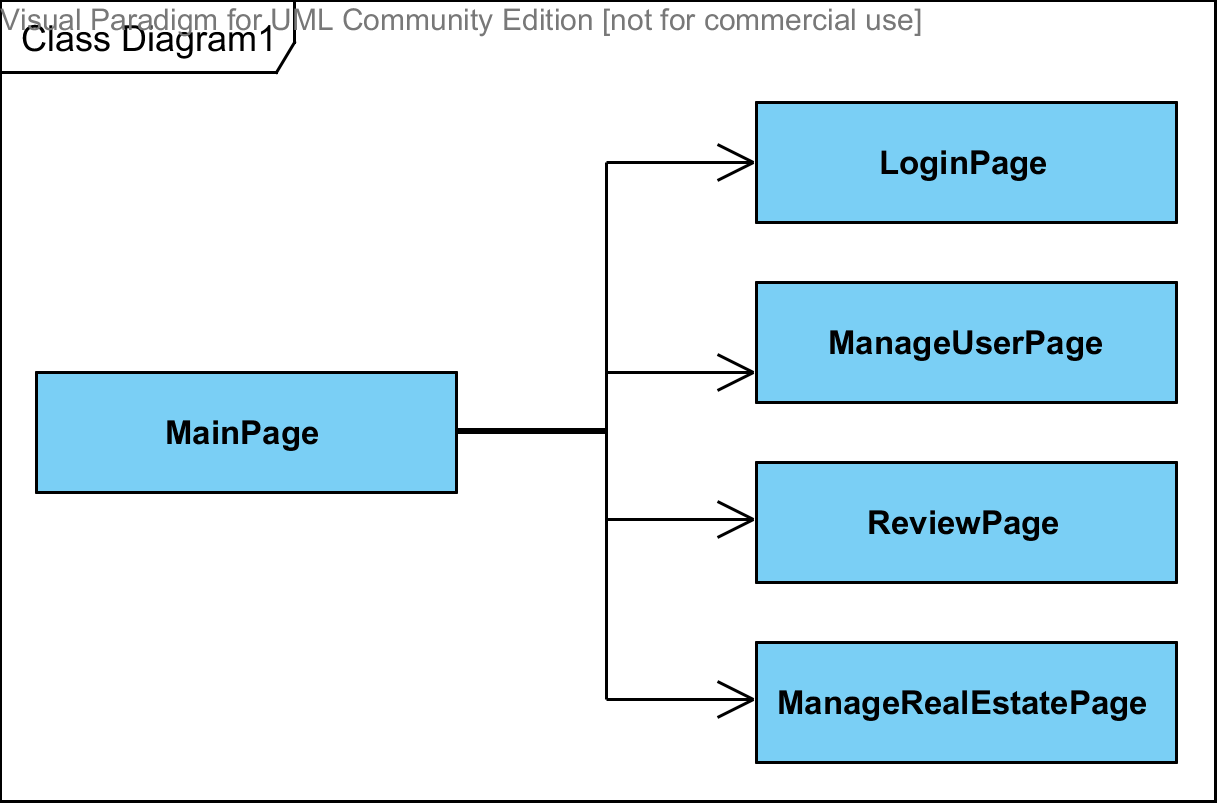
# Tổ chức

* **Nhóm màn hình người dùng**



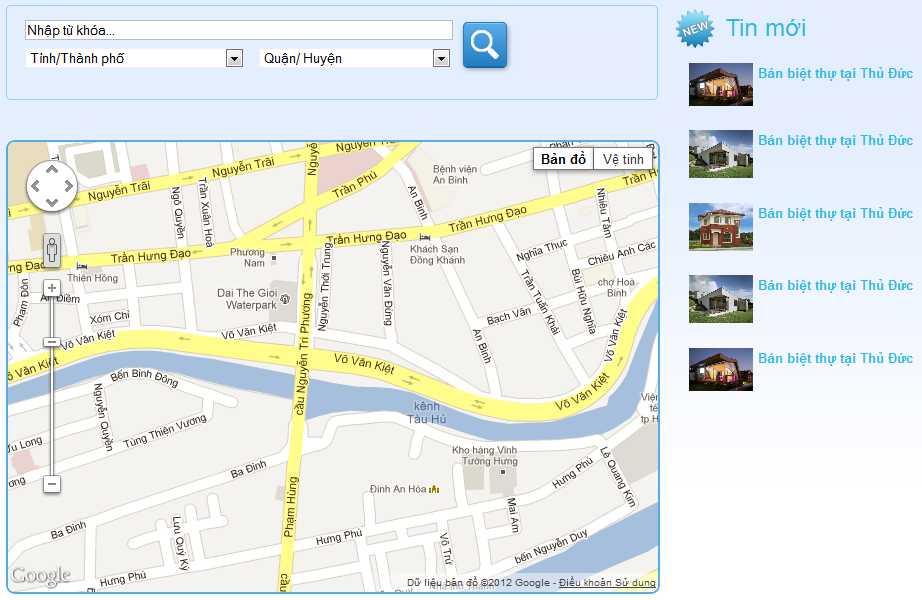
Hình : Sơ đồ màn hình dành cho người dùng

* **Nhóm màn hình quản trị**

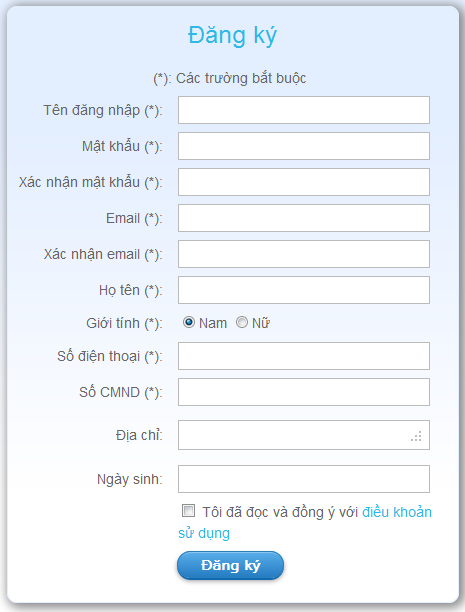
****

Hình : Sơ đồ màn hình dành cho quản trị

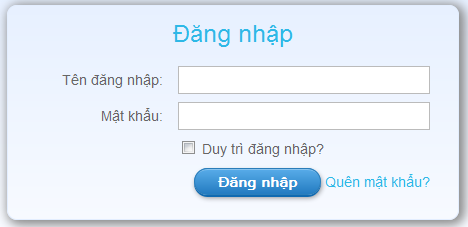
# Danh sách các màn hình



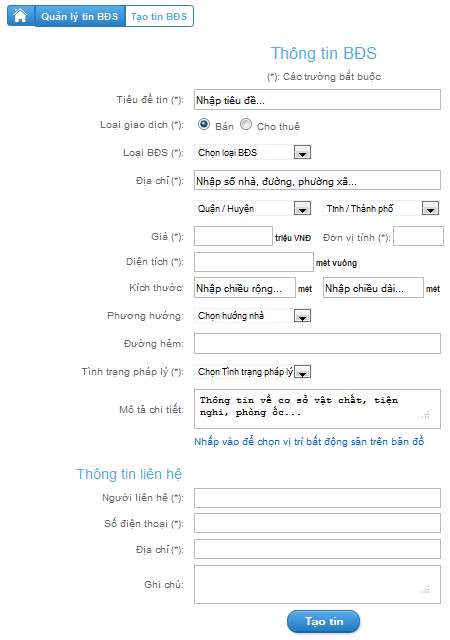
Hình : Trang chủ



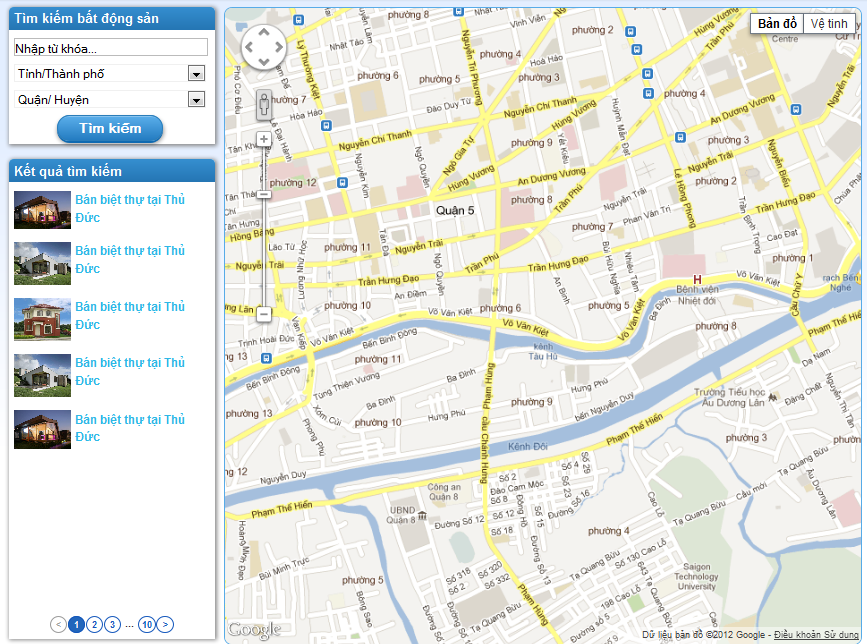
Hình : Màn hình đăng ký



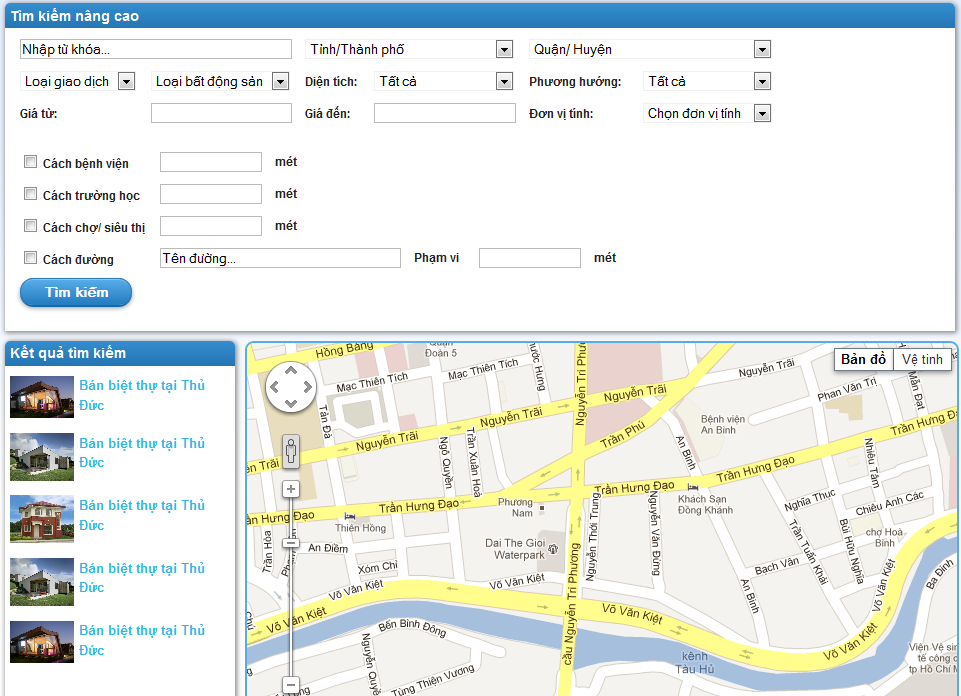
Hình : Màn hình đăng nhập

****

Hình : Màn hình đăng tin bất động sản



Hình : Màn hình tìm kiếm thuộc tính



Hình : Màn hình tìm kiếm không gian nâng cao

# Chương 4: CÀI ĐẶT THỰC NGHIỆM VÀ KIỂM TRA



# Cài đặt

# Môi trường phát triển hệ thống

* Hệ điều hành: Windows.
* Hệ quản trị CSDL: PostgreSQL.
* Công cụ phân tích thiết kế: Visual Paradigm.
* Công cụ xây dựng ứng dụng: Apache, PHP, CodeIgniter Framework, Google Map Service.

# Môi trường triển khai

* **Đối với Server**
  + Hệ điều hành Windows hoặc Linux.
  + Cần cài đặt Apache, PostgreSQL, WampServer.
* **Đối với Client**
  + Hệ điều hành Windows hoặc Linux
  + Kết nối Internet.
  + Trình duyệt hỗ trợ Javascript (Firefox, Chrome).

# Biểu đồ triển khai (Deployment Diagram)



Hình : Sơ đồ triển khai

# Kết quả đạt được

# Chương 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN



# Kết luận

# Hướng phát triển

# PHỤ LỤC A – ĐẶC TẢ USECASE

**A.1 Use case “Đăng ký”:**

* Tóm tắt:
* Use case mô tả việc đăng kí tài khoản của người dùng hệ thống.
* Dòng sự kiện:
  + Use case bắt đầu khi người dùng click vào đường dẫn đăng kí tài khoản trên trang chủ.
  + Dòng sự kiện chính:
    - Hệ thống hiển thị form đăng kí, gồm các thông tin (\* là các trường bắt buộc):
      * Tên đăng nhập (\*)
      * Mật khẩu (\*)
      * Email (\*)
      * Họ tên (\*)
      * Giới tính (\*)
      * Số điện thoại (\*)
      * CMND (\*)
      * Địa chỉ
      * Ngày sinh
    - Người dùng điền các thông tin vào form đăng kí, nhập mã xác nhận và xác nhận đồng ý với điều khoản sử dụng, sau đó nhấn nút “Xác nhận”.
    - Hệ thống thực hiện kiểm tra tính hợp lệ của các thông tin do người dùng cung cấp (định dạng trường thuộc tính, mật khẩu xác nhận, email xác nhận), kiểm tra tên đăng nhập và email đã được sử dụng chưa.
    - Nếu các thông tin do người dùng cung cấp hợp lệ, hệ thống thực hiện mã hóa mật khẩu (sử dụng thuật toán MD5), phát sinh id, sau đó thêm thông tin người dùng vào bảng dữ liệu tài khoản, cùng với thông tin về ngày tạo.
    - Hệ thống thực hiện gửi mail thông báo tạo tài khoản thành công cho người dùng dựa vào thông tin email do người dùng cung cấp.
    - Hệ thống hiện thị thông báo “Tài khoản của bạn đã được tạo thành công!” và chuyển người dùng về trang đăng nhập.
  + Dòng sự kiện khác:
    - Trong luồng sự kiện chính, khi thực hiện kiểm tra tính hợp lệ của thông tin do người dùng cung cấp, nếu thông tin không hợp lệ hệ thống sẽ hiển thị lại form đăng kí cùng các thông báo lỗi tương ứng với các thông tin không hợp lệ.
* Điều kiện tiên quyết:
  + Không có.
* Trạng thái hệ thống sau khi thực hiện use case:
  + Nếu use case thành công, bảng dữ liệu tài khoản được thêm một dòng với các thông tin do người dùng cung cấp. Ngược lại, trạng thái hệ thống không đổi.
* Mở rộng:
  + Không có.

**A.2 Usecase “Đăng nhập”**

* Tóm tắt:
* Use-case mô tả người dùng và quản trị đăng nhập vào hệ thống
* Dòng sự kiện
* Dòng sự kiện chính :

Usecase bắt đầu khi người dùng/quản trị đăng nhập bằng cách chọn đường link trên trang chủ.

* Hệ thống hiện form đăng nhập yêu cầu nhập tên đăng nhập, mật khẩu.
* Người dùng/quản trị nhập tên đăng nhập, mật khẩu, sau đó nhấn “Đăng nhập” hoặc nhấn “Enter”.
* Hệ thống kiểm tra tên đăng nhập, mật khẩu, kiểm tra quyền của tài khoản.
* Hệ thống đăng nhập và chuyển trang tương ứng với quyền của tài khoản và hiện thông báo “Đăng nhập thành công”. Nếu là người dùng, hệ thống sẽ chuyển đến trang quản lý bất động sản. Nếu là quản trị, hệ thống sẽ chuyển đến trang xét duyệt bất động sản.
* Dòng sự kiện khác:
* Khi thông tin đăng nhập không đúng: Hệ thống thông báo “Nhập sai tài khoản hoặc mật khẩu” và hiện form đăng nhập yêu cầu thực hiện lại.
* Điều kiện tiên quyết :
* Không có
* Trạng thái hệ thống sau khi thực hiện use-case:
* Nếu use case thành công, nguời dùng/quản trị đăng nhập vào hệ thống với các quyền được tương ứng. Ngược lại, trạng thái hệ thống không thay đổi.
* Mở rộng :
* Không có

**A.3 Usecase “Quản lý người dùng”**

* Tóm tắt:

Use case này cho phép quản trị quản lý thông tin của các tài khoản người dùng (tài khoản quản trị và tài khoản người dùng thông thường). Use case gồm các thao tác thêm mới, xóa, sửa thông tin trong hệ thống.

* Dòng sự kiện :
* Dòng sự kiện chính

Use-case bắt đầu khi quản trị muốn thêm, xóa, sửa thông tin của tài khoản trong hệ thống.

* + Hệ thống hiển thị danh sách các tài khoản được nhóm thành 2 loại là tài khoản người dùng và tài khoản quản trị. Mặc định sẽ hiển thị danh sách tài khoản quản trị.
  + Hệ thống yêu cầu quản trị chọn một trong các chức năng (thêm, xóa, sửa). Đối với danh sách tài khoản người dùng chỉ có thể thực hiện chức năng Xóa, ngược lại đối với tài khoản quản trị có thể thực hiện đầy đủ 3 chức năng.
  + Sau khi quản trị chọn chức năng, một trong các luồng sau được thực hiện:
  + Nếu quản trị lựa chọn Thêm mới thì dòng sự kiện “Thêm mới tài khoản” được thực hiện.
  + Nếu quản trị lựa chọn Xóa thì dòng sự kiện “Xóa tài khoản” được thực hiện.
  + Nếu quản trị chọn Sửa thì dòng sự kiện “Sửa đổi thông tin tài khoản” được thực hiện.
* **Thêm mới tài khoản:**
* Hệ thống hiện form nhập liệu và yêu cầu quản trị nhập thông tin của tài khoản quản trị.
* Quản trị nhập thông tin về tài khoản mới, thông tin bao gồm.
  + Họ và tên (\*)
  + Ngày Sinh(\*)
  + Địa chỉ liên lạc.
  + Email (\*)
  + Điện thoại.
* Quản trị nhấn “Thêm”.
* Hệ thống kiểm tra thông tin nhập vào (kiểm tra định dạng ngày tháng dd/mm/yyy, email hợp lệ, điện thoại chỉ chứa kí tự số).
* Hệ thống sẽ phát sinh tên đăng nhập và mật khẩu giống nhau cho tài khoản quản trị mới (mật khẫu sẽ mã hóa MD5) và lưu thông tin của tài khoản mới tạo xuống cơ sở dữ liệu.
* Hiển thị danh sách tài khoản mới cập nhập lại lên màn hình, thông báo “thực hiện thành công”, ngược lại nếu hệ thống không thực hiện được thêm mới tài khoản, hiện thông báo “thực hiện không thành công”.
* **Xóa tài khoản:**
* Quản trị chọn tài khoản muốn xóa và nhấn “Xóa”.
* Hệ thống hiển thị thông báo “Bạn thực sự muốn xóa” và yêu cầu quản trị xác nhận muốn xóa tài khoản được chọn hay không.
* Quản trị xác nhận thao tác bắng cách chọn “Đồng ý”.
* Hệ thống xóa tài khoản được chọn ra khỏi cơ sở dữ liệu.
* Hiển thị lại danh sách tài khoản và thông báo “thực hiện thành công”, ngược lại nếu không thể xóa tài khoản, hiện thông báo “Thực hiện không thành công”.
* **Chỉnh sửa:**
* Quản trị chọn tài khoản muốn sửa đổi thông tin bằng cách chọn “Chỉnh sửa” tại dòng chứa tài khoản trên lưới.
* Hệ thống hiển thị thông tin của tài khoản được chọn và hiển thị form nhập liệu đã được điền với thông tin cũ.
* Quản trị sửa đổi các thông tin của tài khoản.
* Sau khi sửa đổi thông tin người dùng chọn “Lưu”.
* Hệ thống kiểm tra thông tin sửa đổi hợp lệ (tương tự trong luồng thêm mới).
* Hệ thống cập nhật thông tin của tài khoản được chọn.
* Hệ thống hiển thị lại danh sách tài khoản cho người dùng và hiển thị thông báo “thực hiện thành công”, ngược lại nếu hệ thống không thể cập nhật, hiện thông báo “thực hiện không thành công”.
* Dòng sự kiện khác:
* **Thông tin về tài khoản không hợp lệ:** Các thông tin được nhập vào trong luồng phụ “Thêm tài khoản” và “Chỉnh sửa” không hợp lệ (Không đầy đủ thông tin cần thiết, username (email) đã tồn tại trong hệ thống,…) thì hệ thống sẽ hiển thị thông báo lỗi và yêu cầu chỉnh sửa.
* **Việc xác nhận không xóa tài khoản:** Hệ thống trở về trạng thái trước đó.
* Điều kiện tiên quyết :
* Quản trị phải đăng nhập vào hệ thống.
* Trang thái hệ thống sau khi thực hiện use-case:
* Nếu use-case thành công, các thao tác thêm, xóa, sửa thông tin tài khoản sẽ được cập nhật. Ngược lại trạng thái của hệ thống không thay đổi.
* Mở rộng :
* Không có

**A.4 Usecase “Quản lý Bất động sản”**

* Tóm tắt:
* Use-case cho phép người dùng/quản trị thực hiện chức năng đăng một tin tức bất động sản mới, xóa hoặc chỉnh sửa thông tin về bất động sản.
* Dòng sự kiện
* Dòng sự kiện chính :

Usecase được thực hiện khi người dùng muốn quản lý các bất động sản đã đăng trong quá trình sử dụng.

* Hệ thống hiển thị danh sách các bất động sản đã đăng và đang chờ xét duyệt.
* Hệ thống hiển thị yêu cầu lựa chọn chức năng : Đăng tin mới, xóa, chỉnh sửa.
  + Sau khi người dùng/quản trị chọn chức năng, một trong các luồng sau được thực hiện:
  + Nếu quản trị lựa chọn Đăng tin mới thì dòng sự kiện “Đăng tin” được thực hiện.
  + Nếu quản trị lựa chọn Xóa thì dòng sự kiện “Xóa tin” được thực hiện.
  + Nếu quản trị chọn Chỉnh sửa thì dòng sự kiện “Chỉnh sửa tin” được thực hiện.
* **Đăng tin:**
* Hệ thống hiện form nhập liệu và yêu cầu người dùng/quản trị nhập thông tin của bất động sản.
* Người dùng/Quản trị nhập thông tin về Bất động sản bao gồm
  + - * Tiêu đề của bất động sản (\*)
      * Địa chỉ bất động sản (\*)
      * Giá cả , đơn vị tính. (\*)
      * Diện tích. (\*)
      * Kích thước.
      * Phương hướng.
      * Đường hẻm.
      * Tình trạng pháp lý. (\*)
      * Loại hình bất động sản. (\*)
      * Loại giao dịch.(\*)
      * Thông tin mô tả: chi tiết về cơ sở vật chất, tiện nghi, v.v...
      * Hình ảnh
* Người dùng nhấn “Vị trí trên bản đồ”.
* Hệ thống hiển thị bản đồ yêu cầu người dùng chọn vị trí của bất động sản trên bản đồ tương ứng.
* Người dùng chọn vị trí của bất động sản trên bản đồ và chọn “Cập nhật vị trí”.
* Sau khi chọn vi trí, người dùng xác nhận thêm mới bằng cách nhấn “Đăng tin”.
* Hệ thống kiểm tra thông tin nhập vào (đầy đủ các thông tin bắt buộc bao gồm cả thông tin vị trí trên bản đồ).
* Hệ thống lưu thông tin của bất động sản mới và chuyển vào danh sách bất động sản chờ xét duyệt sau đó hiển thị thông báo “thực hiện thành công”, ngược lại, nếu hệ thống không thể lưu thông tin bất động sản mới thì hiện thông báo “thực hiện không thành công”.
* **Xóa tin:**
* Người dùng/quản trị chọn bất động sản muốn xóa trên lưới và nhấn “Xóa”
* Hệ thống hiện thông báo “Bạn thực sự muốn xóa” và yêu cầu người dùng xác nhận muốn xóa bất động sản.
* Người dùng/quản trị xác nhận thao tác bằng cách chọn “Đồng ý”.
* Hệ thống xóa bất động sản và hiện thị lại danh sách sau đó hiện thông báo “thực hiện thành công”, ngược lại nếu hệ thống không thể xóa tin (ràng buộc dữ liệu…) thì hiện thông báo “thực hiện không thành công”.
* **Chỉnh sửa tin:**
* Người dùng/quản trị chọn bất động sản muốn chỉnh sửa và nhấn “Chỉnh sửa”.
* Hệ thống hiển thị form nhập liệu được điền bởi thông tin của bất động sản.
* Người dùng/quản trị sửa đổi các thông tin của bất động sản (tương tự trong luồng thêm).
* Hệ thống kiểm tra thông tin sửa đổi.
* Hệ thống cập nhật thông tin của bất động sản được chọn và chuyển vào danh sách bất động sản đang chờ xét duyệt sau đó hiện thông báo “thực hiện thành công”, ngược lại, nếu hệ thống không thể thực hiện cập nhật thông tin bất động sản thì hiện thông báo “thực hiện không thành công”.
* Dòng sự kiện khác:
* **Thông tin về bất động sản không hợp lệ:** Các thông tin được nhập vào trong luồng phụ “Đăng tin” và “Chỉnh sửa tin” không hợp lệ (Không đầy đủ thông tin cần thiết, sai định dạng…) thì hệ thống sẽ hiển thị thông báo lỗi và yêu cầu chỉnh sửa lại.
* **Việc xác nhận không xóa tin:** Hệ thống trở về trạng thái trước đó.
* Điều kiện tiên quyết :
* Người dùng/Quản trị phải đăng nhập vào hệ thống.
* Trạng thái hệ thống sau khi thực hiện use-case:
  + Nếu use-case thành công, các thao tác đăng tin , xóa, sửa bất động sản sẽ được cập nhật. Ngược lại trạng thái của hệ thống không thay đổi.
* Mở rộng :
* Không có

**A.5 Usecase “So sánh bất động sản”**

* Tóm tắt:
* Use-case thực hiện chức năng so sánh các bất động sản do người dùng chọn dựa trên một số tiêu chí cơ bản.
* Dòng sự kiện
* Dòng sự kiện chính :

Usecase được thực hiện khi người dùng cần so sánh thông tin của một số bất động sản.

* Người dùng chọn các bất động sản cần so sánh và thêm vào danh sách bằng cách chọn “Thêm vào danh sách”.
* Người dùng chọn xem lại các bất động sản bằng cách chọn “So sánh”.
* Người dùng có thể bỏ các bất động sản trong danh sách bằng cách chọn vào checkbox trong danh sách bất động sản.
* Hệ thống hiển thị danh sách dạng lưới với các cột là các bất động sản và các dòng lá tiêu chí so sánh. Các bất động sản dựa trên một số tiêu chí so sánh cơ bản. Các tiêu chí so sánh
  + - * Giá cả
      * Diện tích
      * Địa chỉ
      * Tình trạng pháp lý
      * Tiện nghi, tiện ích.
* Điều kiện tiên quyết :
* Không có
* Trạng thái hệ thống sau khi thực hiện use-case:
  + Không có.
* Mở rộng :
* Không có

**A.6 Usecase “Đổi mật khẩu”**

* Tóm tắt:

Use case này cho phép người dùng/quản trị thay đổi mật khẩu.

* Dòng sự kiện:
* Dòng sự kiện chính:

Use case này bắt đầu khi người dùng/quản trị muốn thay đổi mật khẩu

* Người dùng/quản trị chọn “Đổi mật khẩu” trên menu
* Hệ thống hiện form yêu cầu nhập mật : mật khẩu cũ, mật khẩu mới, nhập lại mật khẩu mới, mật khẩu phải >= 6 kí tự.
* Người dùng/quản trị nhập mật khẩu cũ, mật khẩu mới và xác nhận mật khẩu mới.
* Người dùng/quản trị nhấn “Đổi”.
* Hệ thống kiểm tra tính hợp lệ của mật khẩu.
* Hệ thống yêu cầu người dùng/quản trị xác nhận thay đổi mật khẩu bằng cách hiện thông báo “Bạn thực sự muốn thay đổi mật khẩu”.
* Người dùng/quản trị xác nhận thay đổi bằng cách chọn “Đồng ý” trên bảng thông báo.
* Hệ thống cập nhật lại mật khẩu và hiện thông báo “Thực hiện thành công” nếu thực hiện thành công, ngược lại thông báo “Thực hiện không thành công”
* Dòng sự kiện khác:
* Mật khẩu thay đổi không hợp lệ

Mật khẩu không hợp lệ trong các trường hợp sau

+ Nhập sai mật khẩu cũ

+ Mật khẩu mới không phù hợp với số kí tự cho phép.

+ Mật khẩu xác nhận không khớp với mật khẩu mới.

Hệ thống hiện thông báo “mật khẩu thay đổi không hợp lệ” và yêu cầu người dùng/quản trị nhập lại mật khẩu. Người dùng/quản trị có thể chỉnh sửa mật khẩu không hợp lệ hoặc hủy bỏ thao tác đang thực hiện, lúc này use case kết thúc.

* Người dùng/quản trị không xác nhận thay đổi

Nếu người dùng/quản trị không chấp nhận thay đổi thì hệ thống sẽ trở lại trạng thái trước khi thay đổi.

* Các yêu cầu đặc biệt:

Không có

* Điều kiện tiên quyết:

Người dùng/quản trị phải đăng nhập vào hệ thống trước khi use case có thể bắt đầu.

* Trạng thái hệ thống sau khi kết thúc use case:

Nếu use case thành công, mật khẩu vừa thay đổi được cập nhật lại vào hệ thống. Ngược lại, trạng thái của hệ thống không thay đổi.

* Điểm mở rộng:

Không có

**A.7 Use case “Tìm kiếm”**

* Tóm tắt:
  + Use case mô tả việc thực hiện tìm kiếm một mẩu tin bất động sản của người dùng.
* Dòng sự kiện:
  + Use case bắt đầu khi người dùng lựa chọn chức năng tìm kiếm trên thanh menu.
  + Dòng sự kiện chính:
    - Hệ thống hiển thị bản đồ cùng với thông tin vị trí của tất cả các đối tượng bất động sản được đăng trên hệ thống.
      * Màu đỏ: cần bán.
      * Màu vàng: cho thuê.
    - Hệ thống hiển thị form tìm kiếm cơ bản với 3 trường thông tin:
      * Từ khóa
      * Tỉnh thành (hiển thị theo combo box)
      * Quận huyện (hiển thị theo combo box)
    - Nếu người dùng tương tác với bản đồ, dòng sự kiện “Tìm kiếm thông qua tương tác” sẽ được thực thi.
    - Nếu người dùng thực hiện tìm kiếm cơ bản, dòng sự kiện “Tim kiếm cơ bản” sẽ được thực thi.
    - Nếu người dùng nhấn link “Tìm kiếm nâng cao”, dòng sự kiện “Tìm kiếm nâng cao” sẽ được thực thi.
    - **Tìm kiếm thông qua tương tác**
    - Người dùng thực hiện các thao tác trên bản đồ (kéo, phóng to, thu nhỏ).
    - Hệ thống hiển thị lại vị trí các bất động sản dựa theo kích thước và trạng thái hiện tại của bản đồ.
    - Khi người dùng nhấp chuột vào một vị trí bất động sản, hệ thống hiện thị một pop-up trình bày các thông tin:
      * Tiêu đề
      * Địa chỉ
      * Diện tích
      * Giá/đơn vị
      * Loại bất động sản
      * Số điện thoại liên hệ
      * Link “Thông tin chi tiết”
      * Button “Chỉ đường đến đây”
      * Button “Bỏ vào giỏ hàng”
    - **Tìm kiếm cơ bản**
    - Người dùng nhập từ khóa vào ô từ khóa, có thể chọn tỉnh thành và quận huyện từ danh sách, sau đó nhấn nút “Tìm kiếm”.
    - Hệ thống thực hiện tìm kiếm các mẫu tin bằng cách kiểm tra các trường tiêu đề, loại BĐS, địa chỉ, mô tả chi tiết có chứa từ khóa do người dùng cung cấp:
      * Nếu người dùng có lựa chọn tỉnh thành, quận huyện, hệ thống sẽ bổ sung thêm điều kiện vào câu truy vấn.
      * Nếu không hệ thống sẽ lấy thông tin từ tất cả các tỉnh thành, quận huyện.
    - Hệ thống hiển thị kết quả tìm kiếm về cho người dùng dưới dạng danh sách các link đến thông tin chi tiết (được phân trang) với tiêu đề là tiêu đề của mẫu tin.
      * Khi người dùng chuyển trang, hệ thống thực hiện lại câu truy vấn tìm kiếm trước đó và cập nhật lại danh sách trên giao diện.
    - Hệ thống cập nhật lại danh sách các vị trí BĐS trên bản đồ chỉ gồm các vị trí ứng với kết quả tìm kiếm hiện tại.
    - Trên danh sách kết quả, khi người dùng nhấp vào một link kết quả nào đó, sẽ được chuyển con trỏ đến vị trí của BĐS tương ứng cùng pop-up trình bày thông tin.
    - **Tìm kiếm nâng cao**
    - Hệ thống hiển thị form “Tìm kiếm nâng cao” với các trường thông tin:
      * Từ khóa
      * Loại bất động sản (hiển thị theo combo box)
      * Loại giao dịch (hiển thị theo combo box)
      * Tỉnh thành (hiển thị theo combo box)
      * Quận huyện (hiển thị theo combo box)
      * Diện tích (hiển thị theo combo box)
      * Giá
      * Đơn vị tính giá (hiển thị theo combo box)
      * Có trường học trong phạm vi bao nhiêu mét
      * Có bệnh viện trong phạm vi bao nhiêu mét
      * Có chợ/siêu thị trong phạm vi bao nhiêu mét
      * Nằm cách đường bao nhiêu mét (chỉ rõ đường cụ thể)
    - Người dùng nhập từ khóa vào ô từ khóa, lựa chọn các thuộc tính mong muốn từ các trường khác, sau đó nhấn nút “Tìm kiếm”:
    - Hệ thống thực hiện truy vấn các mẩu tin:
      * Hệ thống so khớp từ khóa với trường tiêu đề, mô tả chi tiết
      * Lọc kết quả theo các trường thông tin thuộc tính còn lại
      * Thực hiện các truy vấn không gian.
      * Nếu có dữ liệu trường nào không có giá trị hệ thống sẽ tìm trên tất cả mẩu tin.
    - Hệ thống hiển thị kết quả tìm kiếm về cho người dùng dưới dạng danh sách các link đến thông tin chi tiết (được phân trang) với tiêu đề là tiêu đề của mẫu tin:
      * Khi người dùng chuyển trang, hệ thống thực hiện lại câu truy vấn tìm kiếm trước đó và cập nhật lại danh sách trên giao diện.
    - Hệ thống cập nhật lại danh sách các vị trí BĐS trên bản đồ chỉ gồm các vị trí ứng với kết quả tìm kiếm hiện tại.
    - Trên danh sách kết quả, khi người dùng nhấp vào một link kết quả nào đó, sẽ được chuyển con trỏ đến vị trí của BĐS tương ứng cùng pop-up trình bày thông tin.
  + Dòng sự kiện khác:
    - Trong hai luồng “Tìm kiếm cơ bản” và “Tìm kiếm nâng cao”, nếu không tìm thấy kết quả phù hợp với từ khóa của người dùng cung cấp, hệ thống sẽ hiển thị thông báo “Không tìm thấy kết quả phù hơp với từ khóa được cung cấp!”.
* Điều kiện tiên quyết:
  + Không có.
* Trạng thái hệ thống sau khi thực hiện use case:
  + Không có.
* Mở rộng:
  + Không có

**A.8 Use case “Duyệt bài”**

* Tóm tắt:
  + Use case mô tả việc duyệt các mẩu tin người dùng mới đăng của quản trị hệ thống.
* Dòng sự kiện:
  + Use case bắt đầu khi người dùng chọn chức năng duyệt bài trên trang quản trị hệ thống.
  + Dòng sự kiện chính:
    - Hệ thống hiện thị danh sách các mẩu tin cần phê duyệt, sắp xếp theo trình tự thời gian.
    - Người dùng chọn một mẩu tin cần phê duyệt.
    - Hệ thống hiển thị một pop-up mô tả tất cả thông tin liên quan đến mẩu tin:
      * Tiêu đề của bất động sản
      * Địa chỉ bất động sản
      * Giá cả , đơn vị tính
      * Diện tích
      * Tình trạng pháp lý
      * Loại hình bất động sản
      * Loại giao dịch
      * Thông tin mô tả: chi tiết về cơ sở vật chất, tiện nghi, v.v...
      * Hình ảnh
      * Vị trí trên bản đồ
      * Thông tin liên lạc
      * Người đăng tin
    - Người dùng nhấn nút “Đồng ý đăng tin” hoặc “Từ chối đăng tin” trên cửa sổ pop-up
    - Hệ thống hiển thị khung xác nhận “Bạn có chắc muốn đồng ý/ từ chối đăng tin này?”
    - Người dùng nhấn “Có”
      * Nếu người dùng chọn “Từ chối đăng tin”, hệ thống sẽ hiển thị thêm một khung yêu cầu nhập lý do từ chối, người dùng nhập lý do và nhấn “Xác nhận”.
    - Hệ thống cập nhật lại trạng thái mẩu tin ứng với yêu cầu của người dùng.
    - Hệ thống gửi email thông báo kết quả duyệt tin cho người đăng tin.
      * Nếu tin bị từ chối đăng, hệ thống sẽ ghi thêm lý do từ chối vào mail
      * Nếu tin được đồng ý đăng, hệ thống sẽ gửi kèm link đến mẩu tin trong email.
  + Dòng sự kiện khác:
    - Trong dòng sự kiện chính, khi hiện thị pop-up thông tin chi tiết mẩu tin, nếu người dùng đóng khung pop-up, hệ thống sẽ hiển thị lại danh sách mẩu tin cần duyệt.
    - Trong dòng sự kiện chính, khi hiển thi form xác nhận, nếu người dùng nhấn “Không” hệ thống sẽ hiển thị lại pop-up thông tin chi tiết của mẩu tin.
* Điều kiện tiên quyết:
  + Người dùng phải đăng nhập với quyền admin.
* Trạng thái hệ thống sau khi thực hiện use case:
  + Nếu use case thành công, giá trị trạng thái của mẩu tin trong cơ sở dữ liệu sẽ được cập nhật lại.
* Mở rộng:
  + Không có.

**A.9 Use case “Chỉ dẫn đường đi”**

* Tóm tắt:
  + Use case mô tả việc tìm kiếm đường đi từ một địa điểm bất kì đến vị trí bất động sản của người dùng.
* Dòng sự kiện:
  + Use case bắt đầu khi người dùng nhấn “Tìm đường đến đây” khi nhấp chọn mẩu tin trên bản đồ.
  + Người dùng nhấp chuột phải trên một vị trí bất kì trên bản đồ và chọn “Tìm đường từ đây”
  + Hệ thống gửi thông tin tọa độ hai vị trí được chọn đến Google Map và hiển thị thông tin hướng dẫn đường đi cho người dùng.
* Điều kiện tiên quyết:
  + Người dùng phải ở màn hình tìm kiếm hoặc ở màn hình chi tiết mẩu tin (có hiện thị bản đồ tương tác)
* Trạng thái hệ thống sau khi thực hiện use case:
  + Không có.
* Mở rộng:
  + Không có.

# TÀI LIỆU THAM KHẢO