|  |
| --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  **TRƯỜNG ĐẠI HỌC NÔNG LÂM TP HCM**  **KHOA CÔNG NGHỆ THÔNG TIN**    **BÁO CÁO GIỮA KỲ**  **Đề tài: Hệ thống đặt nhà hàng online**  **Giáo viên hướng dẫn:** Nguyễn Thị Minh Hương  **Thực hiện:** Nhóm 1  **Thành viên:**   * 17130117 - Phan Văn Luận * 17130120 - Lý Đức Mạnh * 17130002 - Ngô Vĩ An * 17130254 - Võ Thanh Trí * 17130181 - Phạm Văn Quang * 17130243 - Lê Văn Tiến   TP. Hồ Chí Minh, Tháng 12 năm 2019 |

1. **Thông tin các thành viên (Team Information).** 
   1. **Danh sách.**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **MSSV** | **HỌ VÀ TÊN** | **CHỨC VỤ** |
| 1 | 17130117 | Phan Văn Luận | *Leader* |
| 2 | 17130120 | Lý Đức Mạnh | Member |
| 3 | 17130002 | Ngô Vĩ An | Member |
| 4 | 17130254 | Võ Thanh Trí | Member |
| 5 | 17130243 | Lê Văn Tiến | Member |
| 6 | 17130181 | Phạm Văn Quang | Member |

* 1. **Vai trò công việc từng thành viên.** 
     1. **Phan Văn Luận.**

**Thiết kế (Activity Diagram, Sequence Diagram):**

* Chức năng quản lý nhà hàng.
* Chức năng quản lý tài khoản.

**Back-end:**

* Thiết kế Database cho chức năng quản lý nhà hàng.
* Thiết kế database chức năng quản lý tài khoản.
  + 1. **Lý Đức Mạnh.**

**Thiết kế (Activity Diagram, Sequence Diagram):**

* Chức năng quản lý review khách hàng.
* Chức năng quản lý liên hệ.

**Back-end:**

* Thiết kế Database cho chức năng quản lý review khách hàng, liên hệ.

**Front-end:**

* Thiết kế giao diện quản lý review, liên hệ.
  + 1. **Ngô Vĩ An.**

**Thiết kế (Activity Diagram, Sequence Diagram):**

* + - * Chức năng quản lý tin tức.
      * Chức năng quản lý lịch đặt chỗ.

**Back-end:**

* Thiết kế Database cho chức năng quản lý tin tức.
* Thiết kế Database cho chức năng quản lý lịch đặt chỗ.  
  + 1. **Võ Thanh Trí.**

**Thiết kế (Activity Diagram, Sequence Diagram):**

* Chức năng quản lý thông tin cá nhân.
* **Front-end:**
* Thiết kế giao diện cho chức năng quản lý thông tin cá nhân.
* Thiết kế giao diện cho chức năng lịch đặt chỗ (phía khách hàng).
* Thiết kế giao diện quản lý lịch đặt chỗ (phía chủ nhà hàng).
  + 1. **Phạm Văn Quang.**
* **Thiết kế (Activity Diagram, Sequence Diagram):**
* Chức năng đặt chỗ nhà hàng.
* **Front-end:**
* Thiết kế giao diện danh sách, chi tiết nhà hàng.
* Thiết kế giao diện cho chức năng xem tin tức, review.
* Thiết kế giao diện cho chức năng đặt chỗ.
  + 1. **Lê Văn Tiến.**
* **Thiết kế (Activity Diagram, Sequence Diagram):**
* Chức năng tìm kiếm.
* Chức năng đăng nhập, đăng ký.
* **Front-end:**
* Thiết kế giao diện cho chức năng đăng ký, đăng nhập.
* Thiết kế giao diện chức năng liên hệ .
* Thiết kế giao diện chức năng quản lý nhà hàng, tài khoản.

1. **Vấn đề cần báo cáo (Problem statement).** 
   1. **Thực tiễn.**

Theo nhận định của các chủ nhà hàng đến thời điểm hiện tại việc đặt bàn vẫn còn rất nhiều hạn chế:

* Mất quá nhiều thời gian để tìm một bàn phù hợp gần nơi sinh sống, nơi mới đến.
* Không xác định được tình trạng bàn còn trống hay đã hết chổ dẫn đến việc mất quá nhiều thời gian đi lại chỉ để tìm một nhà hàng, đối với những người bận rộn thì điều đó là một sự cản trở lớn.
* Không xác định được giá cả phù hợp trước khi đến nhà hàng.
* Chủ nhà hàng không tận dụng hết lợi thế của internet và công nghệ, không xử lý được lượng khách hàng lớn cùng lúc.

Vì thế chúng tôi muốn phát triển hệ thống này để khắc phục được những vấn đề nêu trên và tạo mọi thuận lợi cho người dùng trên mọi góc cạnh chủ nhà hàng hay khách hàng…

* 1. **Mục đích của đồ án (Purpose).**
* Phát triển hệ thống đặt bàn phục vụ cho nhu cầu ăn uống trên toàn quốc.
* Để việc đặt bàn được dễ dàng, nhanh gọn.
* Khách hàng được tiếp cận nhanh với việc đặt bàn, lựa chọn được thời gian thích hợp và nhà hàng ưa thích, trong khu vực sinh sống.
* Tránh được việc chờ đợi và mất thời gian khi nhà hàng không còn chỗ trống.
* Thông tin đặt lịch được xử lý chính xác hơn.
* Chủ nhà hàng có thể khắc phục được việc mất quá nhiều thời gian khi khách hàng đặt bàn nhưng không tới hoặc tới trể. Từ đó dẫn tới việc thu lại lợi nhuận cao cho chủ nhà hàng.
  1. **Phạm vi sử dụng của hệ thống (Scope).**

Mô hình này có thể triển khai trên phạm vi toàn quốc.

Chủ nhà hàng ở khắp nơi có thể apply vào hệ thống sử dụng hình thức đặt bàn nhà hàng online.

1. **Yêu cầu kỹ thuật (Software Requirements Specification).** 
   1. **Các định nghĩa và từ viết tắt (Definition, Acronyms and Abbreviations).** 
      1. **Các định nghĩa.**

**Use-case UML:** Là một sơ đồ thể hiện cách những user trong system có thể tương tác với system bằng những feature gì? Nó sẽ bao gồm các thành phần chính:

* Những loại user trong hệ thống (Actors).
* Những feature có trong hệ thống (Use cases).
* Mối quan hệ giữa Actors và Use cases (Associations).
* Phạm vi của hệ thống của bạn (System boundary boxes-optional).

**Activity diagram:** Là một mô hình logic được dùng để mô hình hóa cho các hoạt động trong một qui trình nghiệm vụ. Nó chỉ ra luồng đi từ hoạt động này sang hoạt động khác trong một hệ thống. Nó đặc biệt quan trọng trong việc xây dựng mô hình chức năng của hệ thống và nhấn mạnh tới việc chuyển đổi quyền kiểm soát trong các đối tượng.

**Back-end:** Là thuật ngữ chỉ các giai đoạn bắt đầu và kết thúc của một quá trình xử lí. Trong thiết kế phần mềm back-end (phần lập trình trên server) gồm có các thành phần xử lí thông tin từ Font-end. Thông thường ám chỉ tới việc tương tác với DBMS (hệ quản trị dữ liệu).

**Front-end:** Tương tác trực tiếp với người sử dụng. Cụ thể đó là hệ thống các giao diện người dùng (GUI) và lập trình phía người dùng.

**Performance requirement:** Là những tiêu chí mà hoàn toàn không thể thương lượng do nghĩa vụ hợp đồng, thỏa thuận cấp độ dich vụ (SLA), hoặc các nhu cầu kinh doanh cố định. Bất kỳ tiêu chí hiệu suất nào mà sẽ không phải là nghi ngờ hàng đầu dẫn đến một quyết định hoãn phát hành cho đến

các tiêu chí đi không hoàn toàn cần thiết và do đó, đây không phải là một yêu cầu.

* + 1. **Từ viết tắt.**

**DB:** Database.

**DBMS:** Database Management System.

**GUI:** Graphical User Interface.

**SLA:** Service Level Agreement.

* 1. **Công nghệ sử dụng (Technologies to be used).** 
     1. **Thiết kế hệ thống.**

 StarUML.

* + 1. **Front-end.**

Bootstrap v4.0.0-beta.2.

FontAwesome.

jQuery 4.0.0-alpha

* + 1. **Back-end.**

 Sử dụng java – servlet.

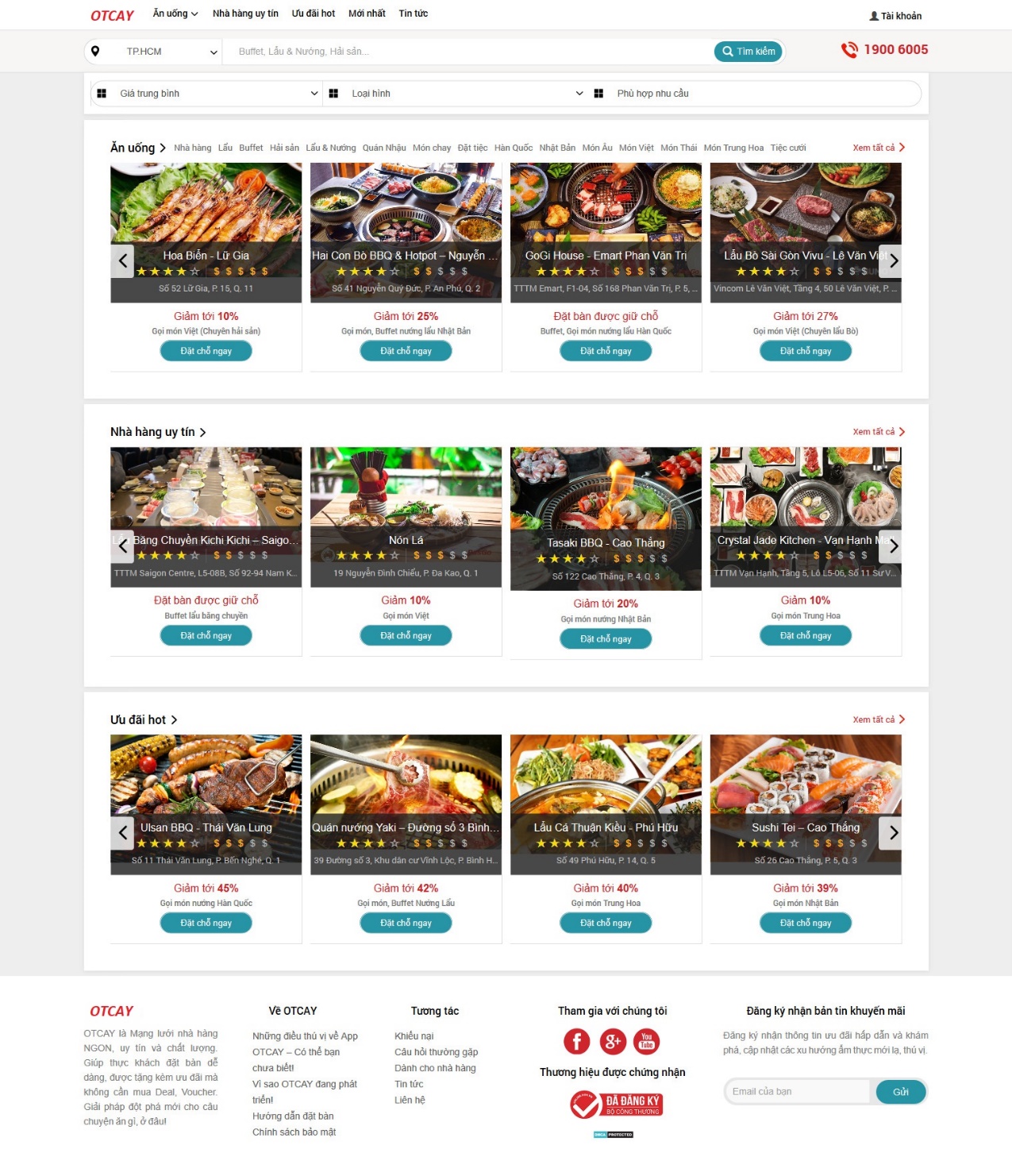
 Ajax(asynchronoys JavaScript & XML).

 Ckeditor, Ckfinder.

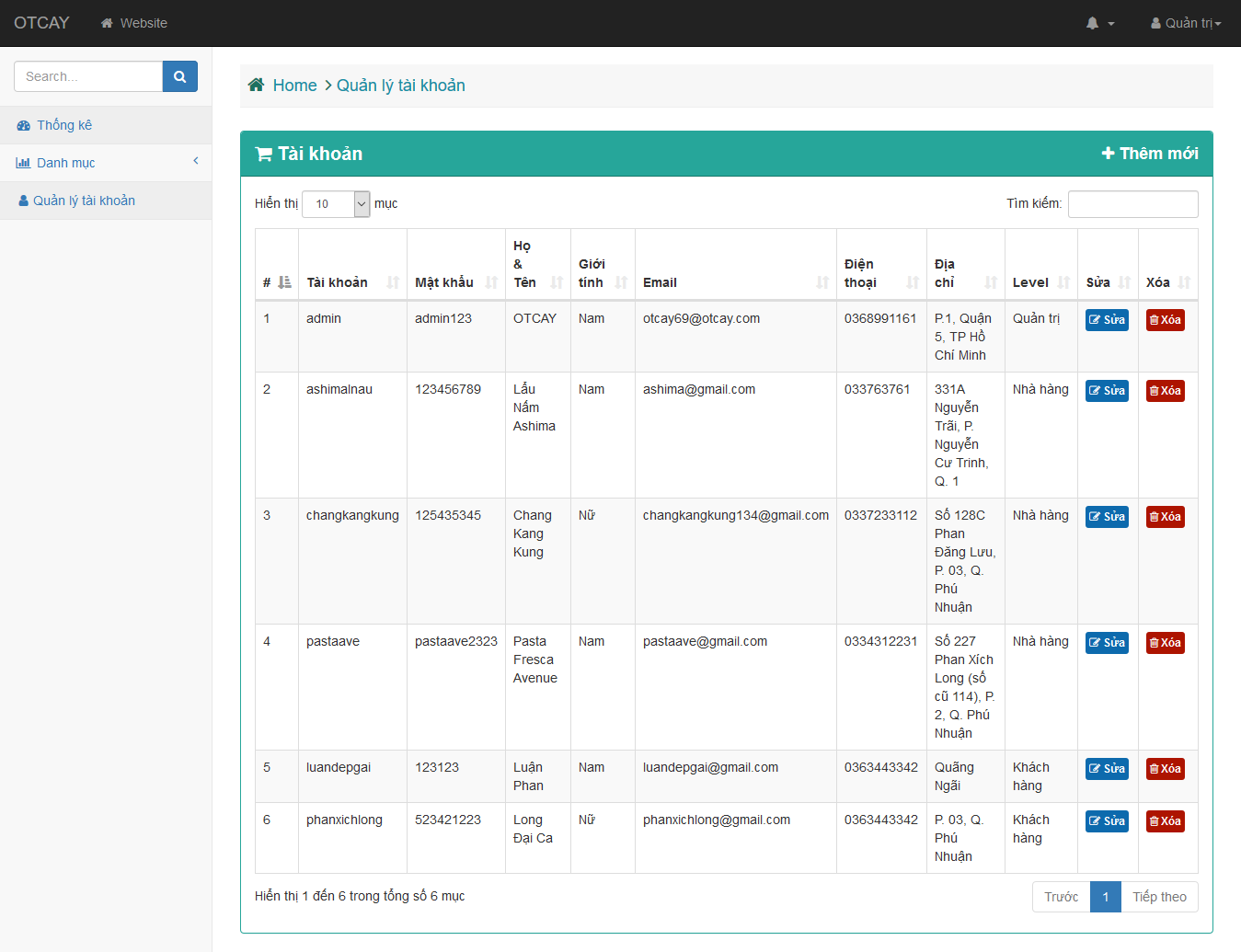
* + 1. **Công cụ đo thời gian đáp ứng.**

 WebPageTest.

* 1. **Giao diện phần mềm (Software interface).** 
     1. **Giao diện Front End Client.**



* + 1. **Giao diện Back-end.**



* + 1. **Mô tả của giao diện phân mềm.**
  1. **Chức năng hệ thống (System functions).** 
     1. **Admin.**

Đăng nhập.

Quản lý danh sách nhà hàng.

Quản lý tài khoản chủ nhà hàng, khách hàng.

Quản lý tin tức.

Quản lý review khách hàng.

Liên hệ.

* + 1. **User khách hàng.**

Xem nhà hàng.

Xem tin tức.

Review.

Tìm kiếm.

Đăng nhập.

Đăng ký.

Đặt chỗ.

Quản lý lịch sử đặt chỗ.

Cập nhật thông tin tài khoản

* + 1. **User chủ nhà hàng.**

Quản lý thông tin nhà hàng.

Quản lý lịch đặt chỗ.

Đăng nhập.

Quản lý review khách hàng.

* 1. **Xác định người dùng hệ thống (User characteristics).** 
     + - * Khách hàng là chủ nhà hàng có nhu cầu muốn quảng bá thông tin nhà hàng.
         * Khách hàng có nhu cầu sử dụng hệ thống với mục đích tìm và đặt lịch nhà hàng.
         * Hình thức đặt bàn qua hệ thống từ xa qua internet.
         * Admin là người quản trị hệ thống.
  2. **Yêu cầu thực hiện (Performance Requirements).** 
     1. **Thời gian đáp ứng.**

 Thời gian đáp ứng sẽ được đo bằng cách sử dụng *WebPageTest*. Bộ hẹn giờ sẽ đo thời gian từ yêu cầu cho một trang đến khi bit cuối cùng được yêu cầu để hiển thị trang được trả lại. Thời gian phản hồi phụ trợ sẽ được đo bằng cách sử dụng tệp nhật ký máy chủ ứng dụng.

* + 1. **Khối lượng công việc.**

 Vào những ngày bận rộn phần mềm có thể hổ trợ được khoảng 5000 lượt truy cập và tạo ra khoảng 1000 lượt tương tác khách hàng.

* + 1. **Khả năng mở rộng.**

 Chúng tôi đảm bảo phần mềm đặt bàn online sẽ có khả năng hỗ trợ ít nhất 5000 lượt truy cập và tạo ra khoảng 1000 lượt đặt chỗ.

* 1. **Yêu cầu phần mềm (Software requirements).** 
     1. **Độ tin cậy (Reliability):**

 Phải thực hiện được chính xác, đầy đủ yêu cầu của người sử dụng.

 Phải có khả năng xử lý và chịu lỗi.

 Phải có khả năng phục hồi, quản lý tài nguyên.

 Quản lý tính toàn vẹn và tính thống nhất của dữ liệu.

 Đảm bảo các yêu cầu về bảo mật.

* + 1. **Bảo mật (Security):**

 Bảo vệ dữ liệu không bị lộ ra ngoài một cách trái phép.

 Chỉ những người dùng được ủy quyền mới được phép chỉnh sửa dữ liệu.

 Đảm bảo dữ liệu luôn sẵn sàng khi những người dùng hoặc ứng dụng được ủy quyền yêu cầu.

 Khả năng ngăn chặn việc từ chối một hành vi đã làm.

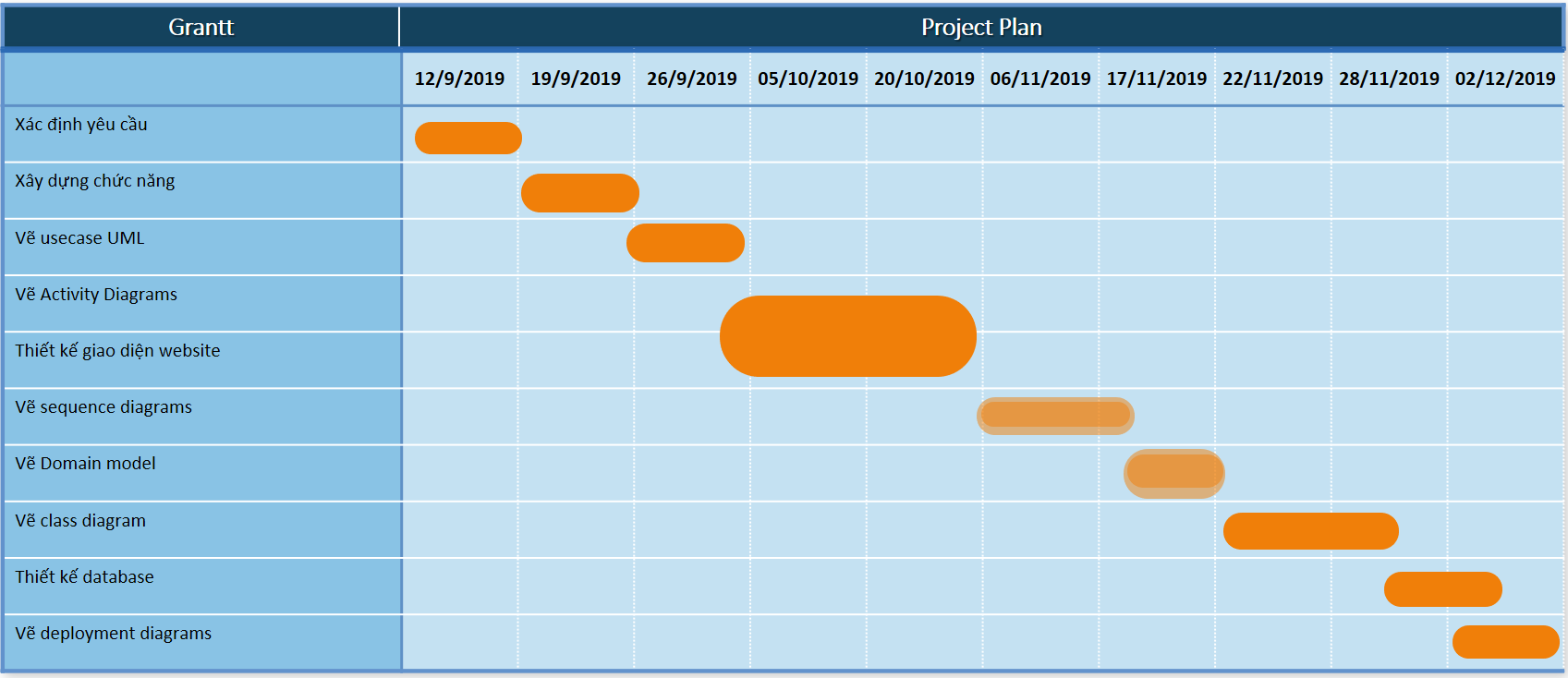
* + 1. **Bảo trì (Maintainability):**

 Bảo trì theo định kì (1 quý/ lần, 2 quý/ lần...) phát hiện sớm các lỗ hổng bảo mật cũng như các mối nguy hại cho hệ thống mạng sẽ giúp doanh nghiệp chủ động đề phòng và khắc phục...

 Khi hệ thống xảy ra lỗi, sự cố cần khắc phục trong quá trình hoạt động.

 Khi người dùng phát sinh thêm chức năng, bỏ bớt hay sửa chữa những chức năng đối với hệ thống.

* + 1. **Portability:** Hệ thống có thể chạy tốt trong nhiều hệ điều hành khác nhau như Window, Mac OS, IOS, Android.

1. **Kế hoạch cho đồ án (Project plan).** 
   1. **Lập tiến độ thực hiện đồ án bằng biểu đồ Grantt.**

* 1. **Xác định các mốc cho các phiên bản của hệ thống.**

 **Trước khi hoàn thành hệ thống:**

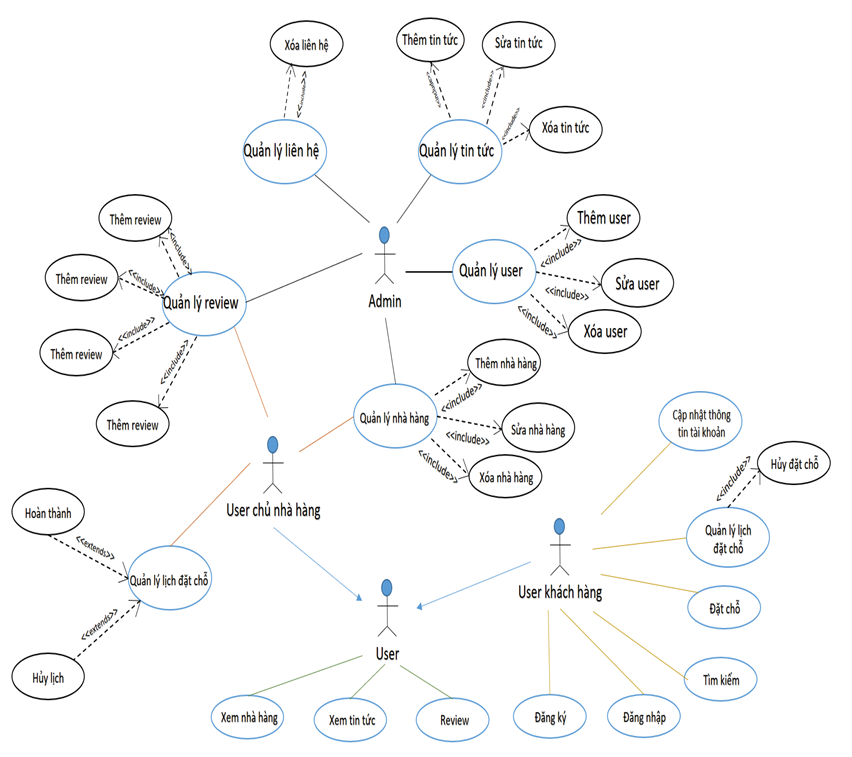
 Dựa vào mức độ hoàn thành của hệ thống, cứ trong vòng 2 tuần làm việc thì sẽ cập nhật 1 version. Để tránh việc mất dữ liệu làm chậm tiến độ của đồ án.

 **Sau khi hoàn thành hệ thống:**

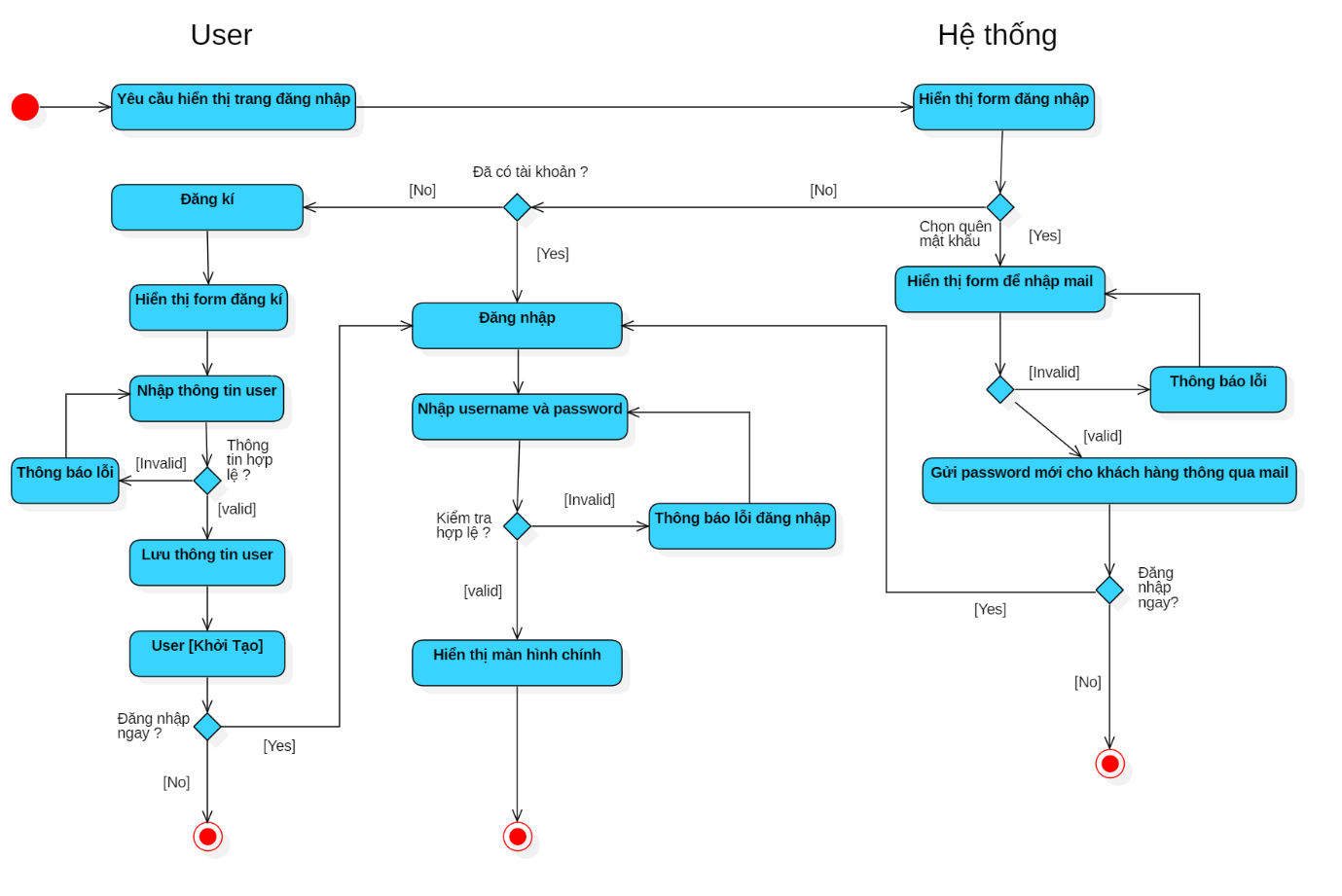
 Mốc cho các phiên bản của hệ thống dựa vào việc bảo trì theo định kì 3 tháng 1 lần, 6 tháng 1 lần.

 Khi khách hàng thay đổi chức năng mới cho hệ thống.

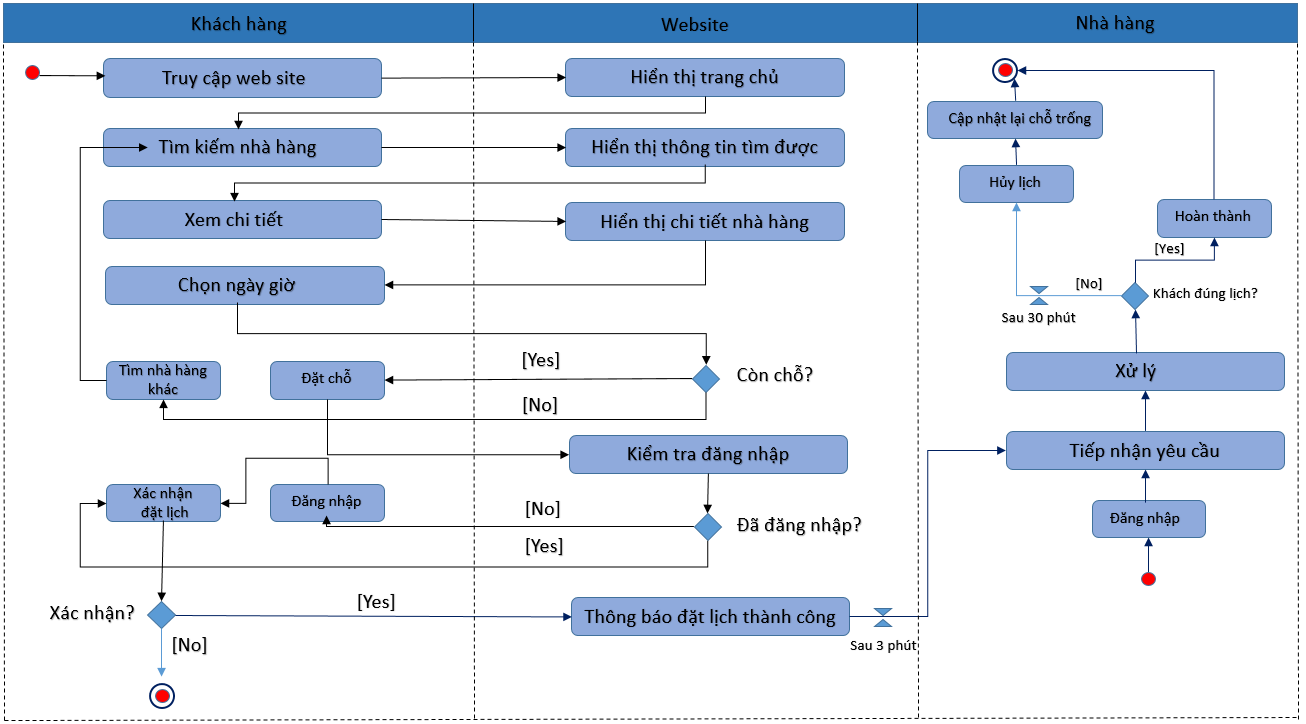
1. **Mô hình hóa yêu cầu:** Use case model, Activity diagram.
   1. **Use case model.**

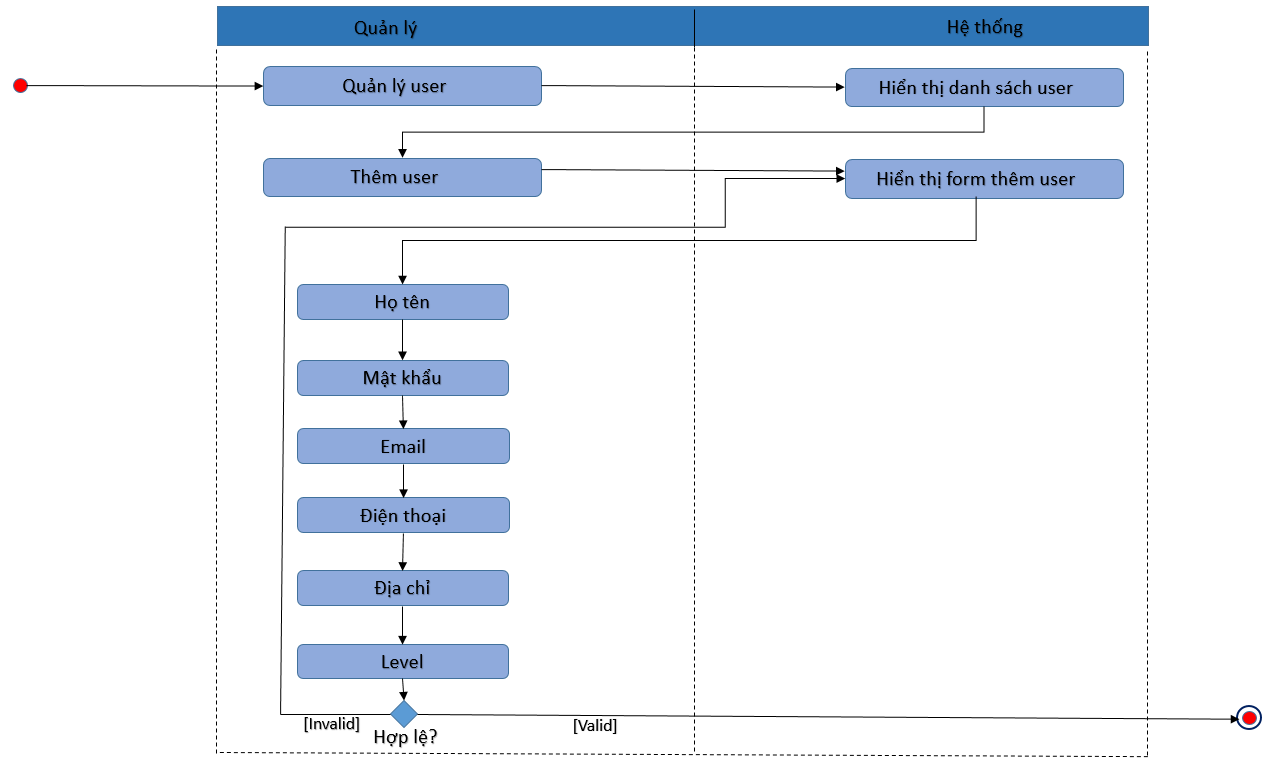
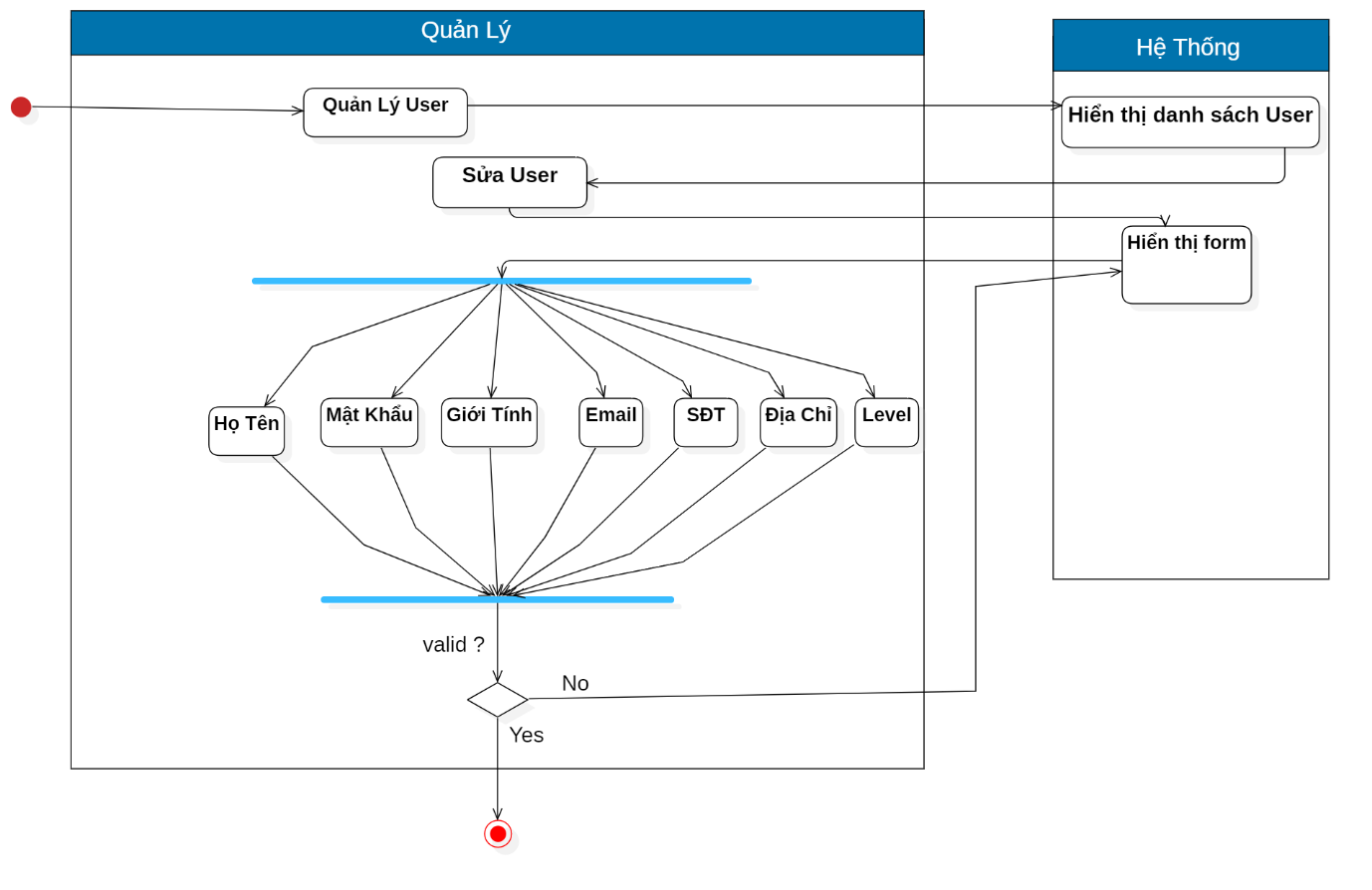
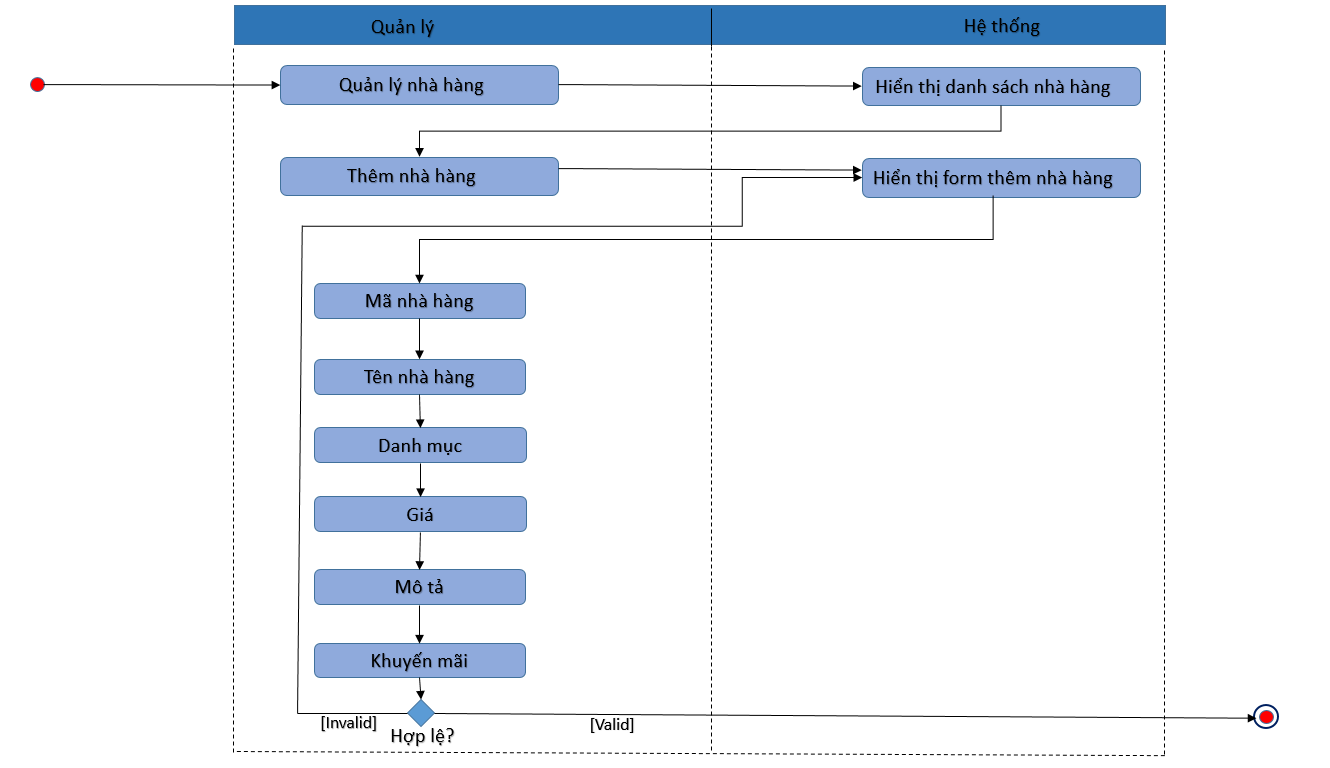
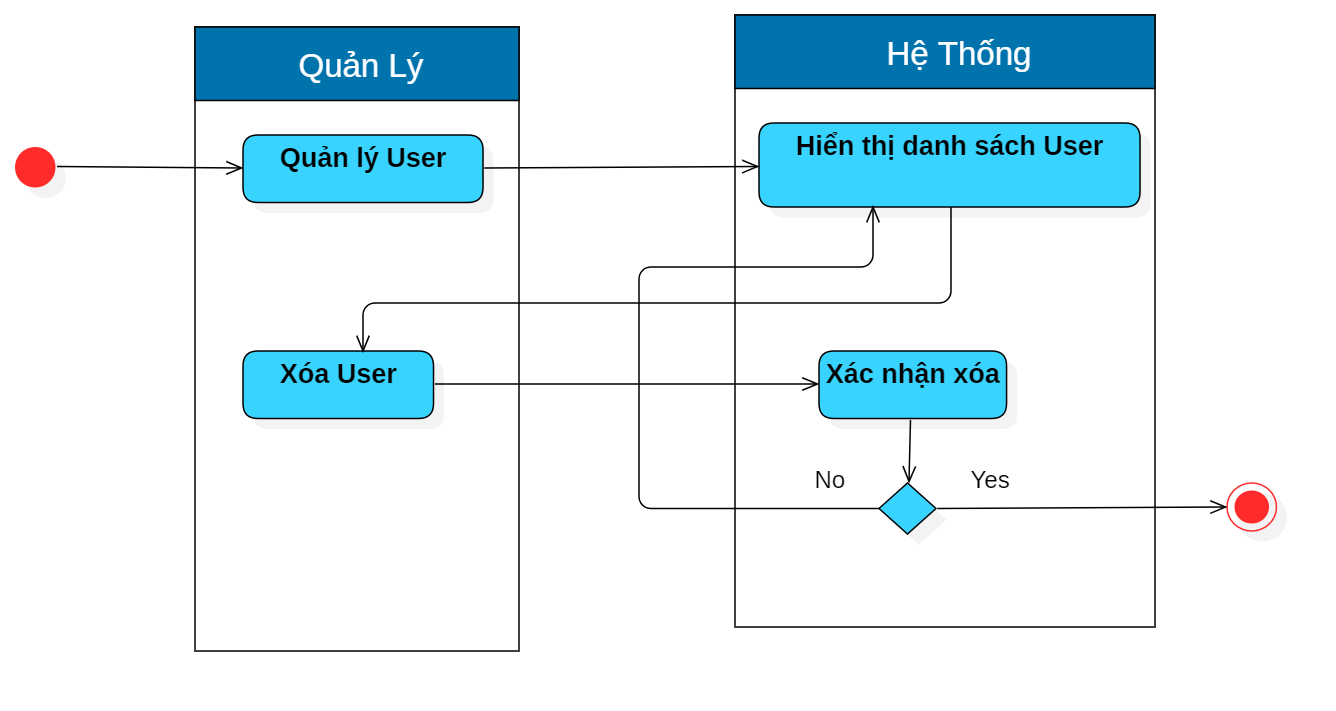


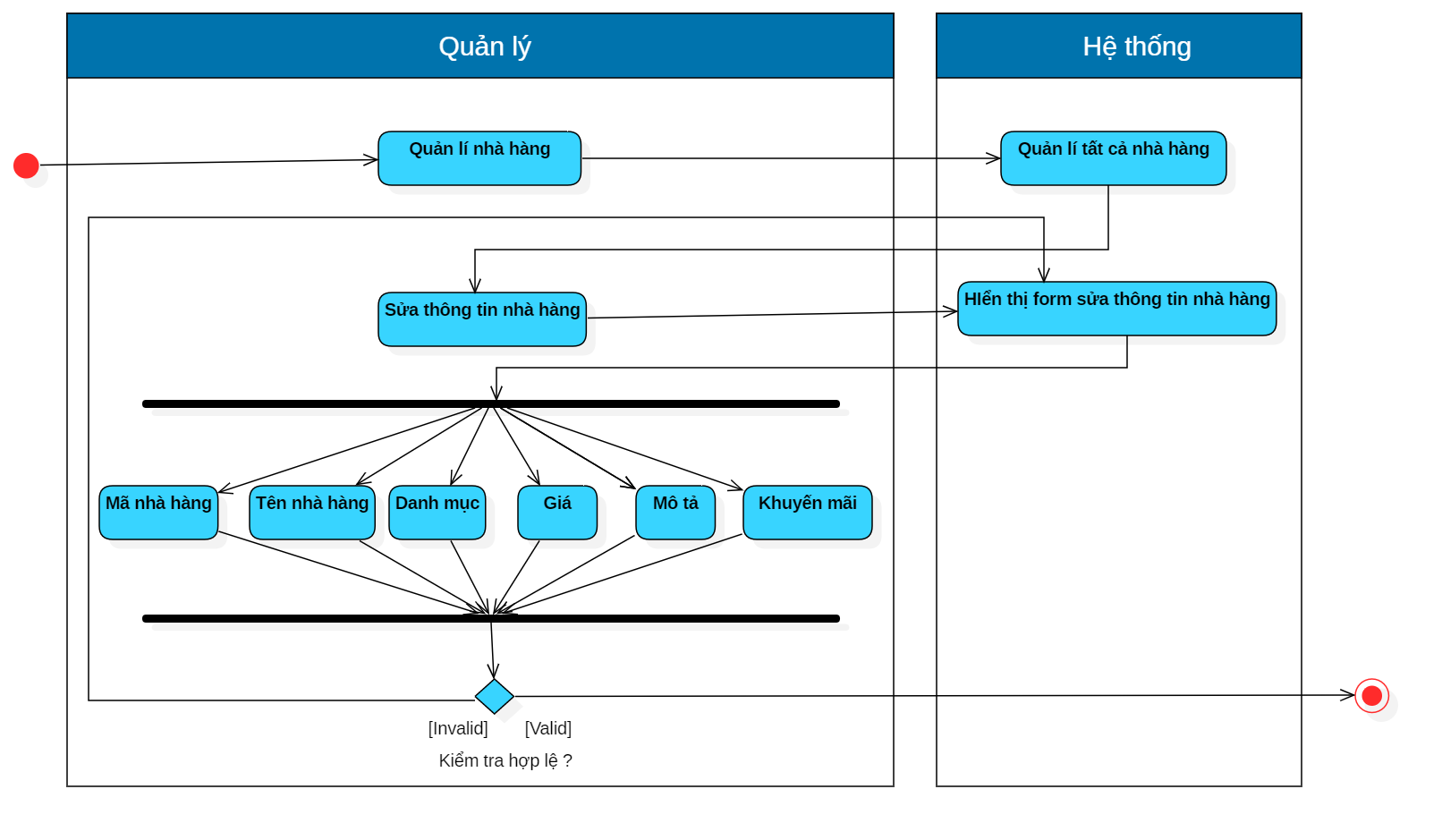
* 1. **Activity diagram.**
     1. **Chức năng đăng nhập, đăng ký.**

****

* + 1. **Chức năng đặt chỗ.**

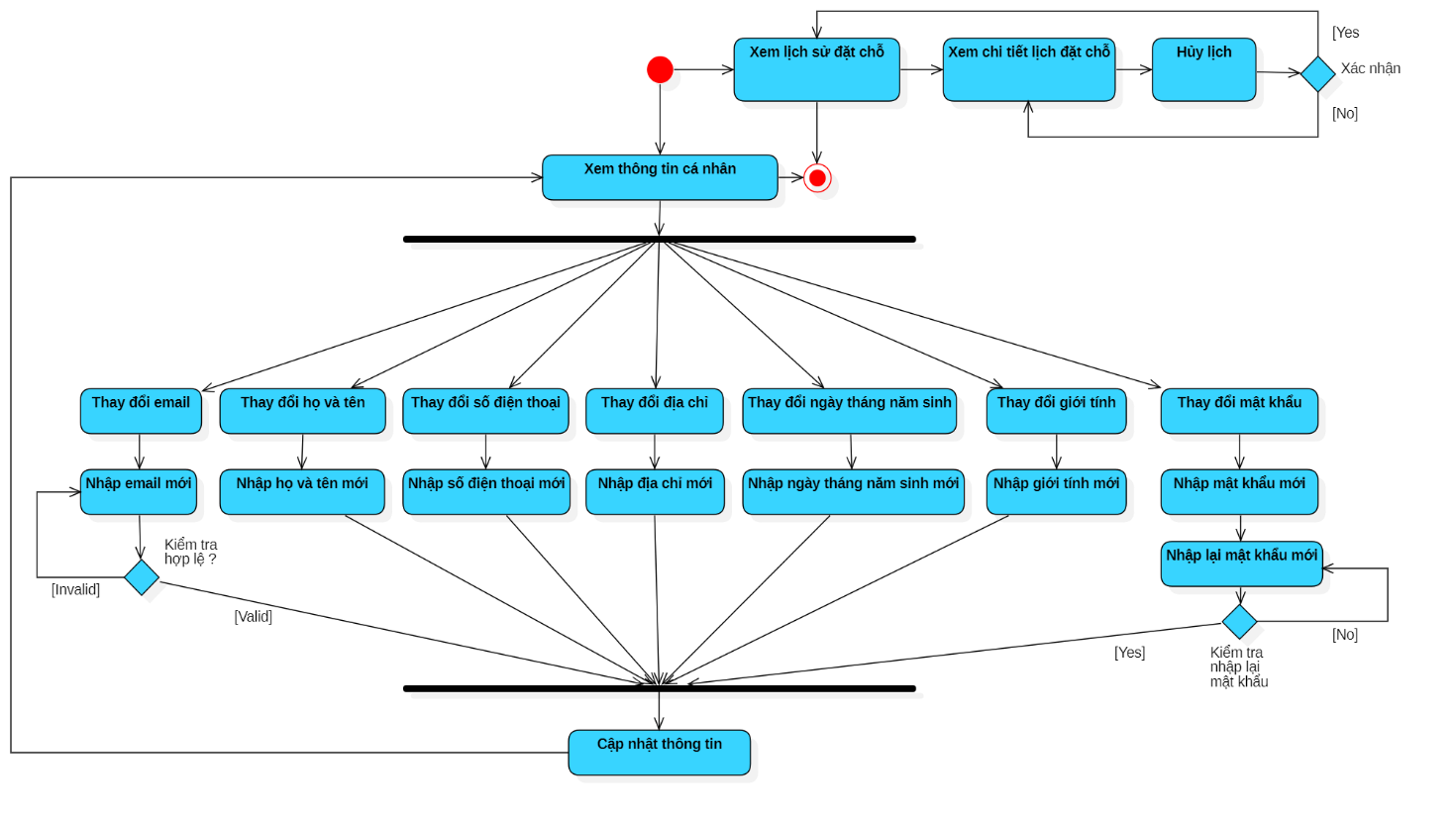


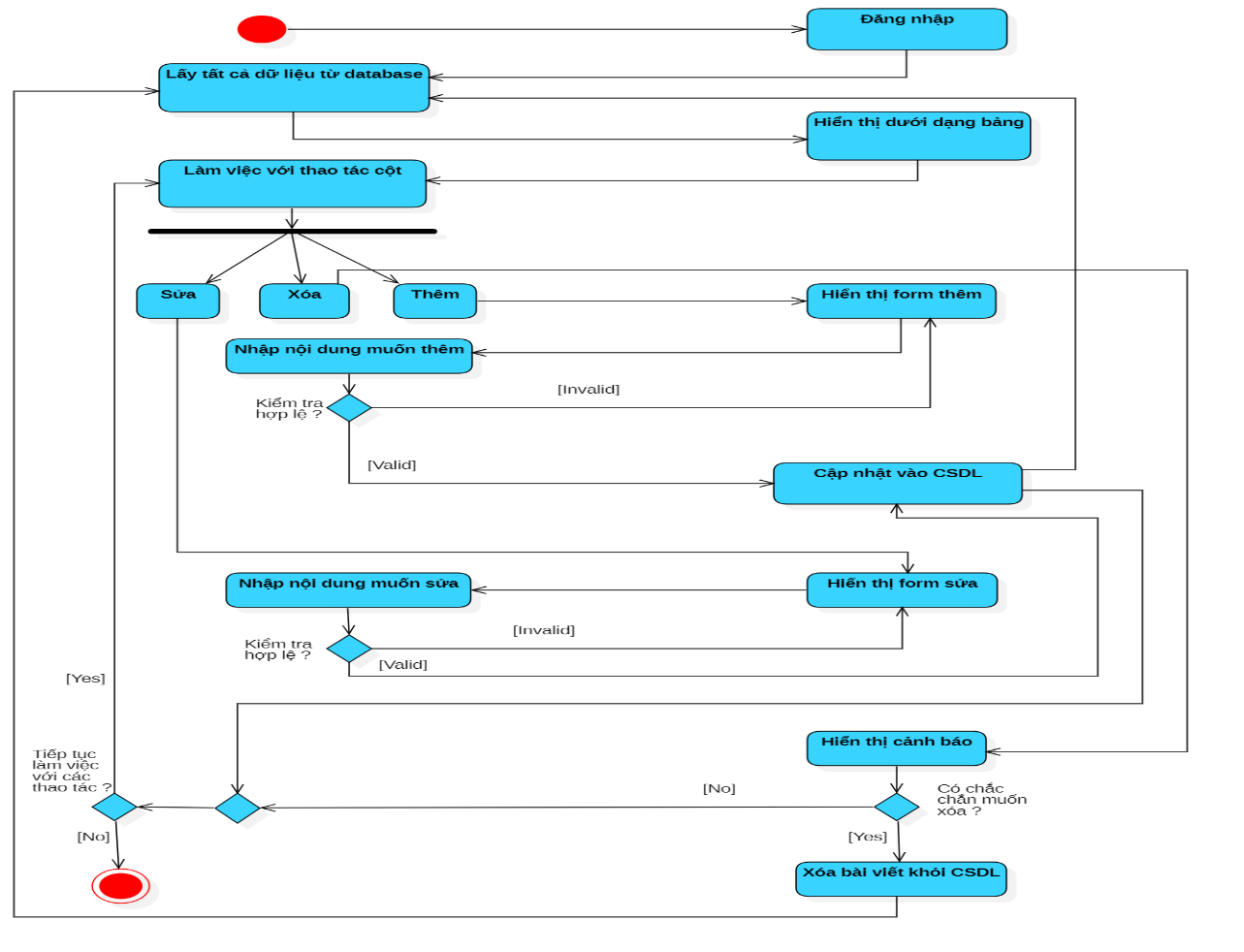
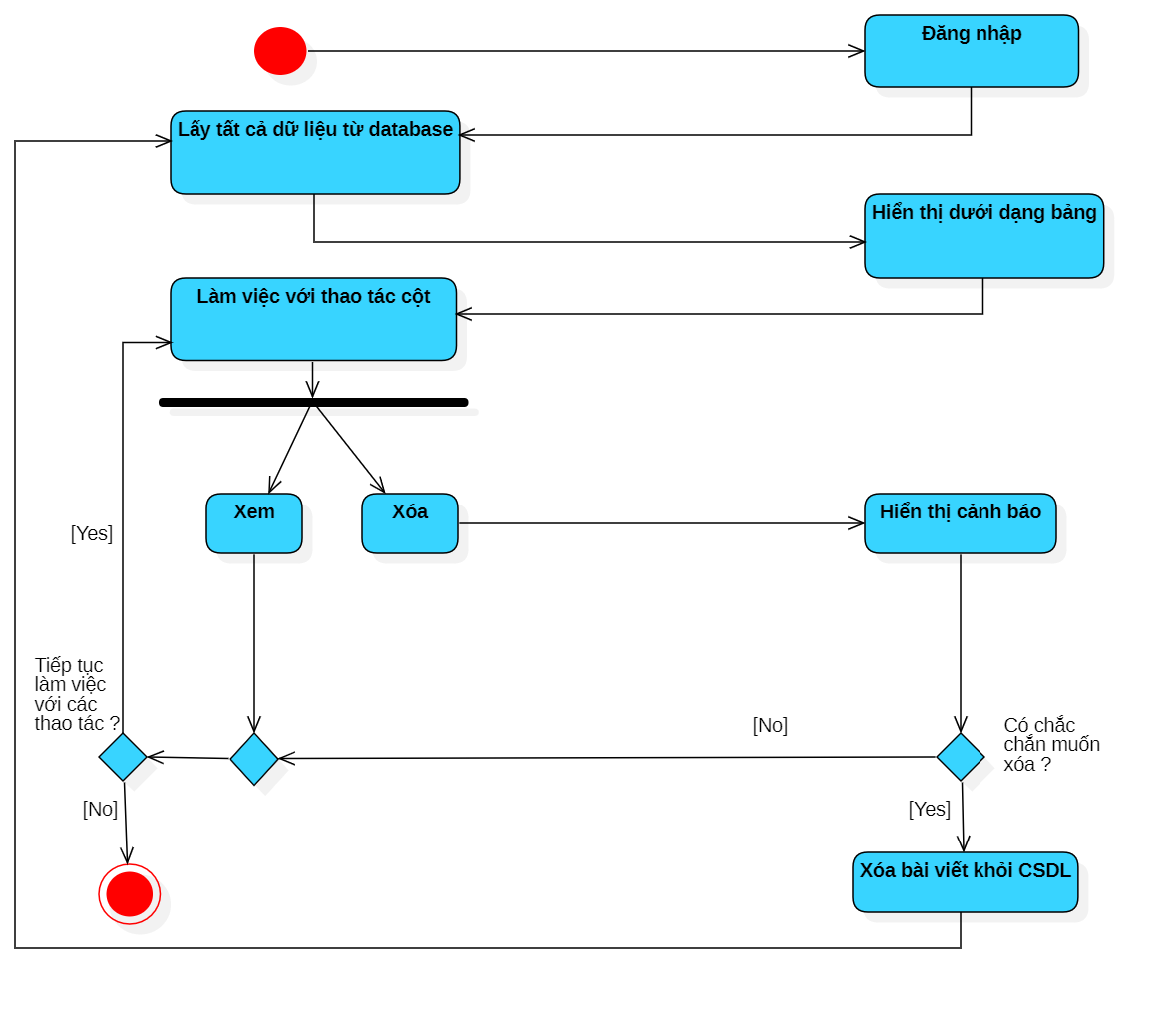
* + 1. **Quản lý user.  
         
         
         
         
         
         
         
         
         
         
         
         
         
         
         
         
         
         
         
         
         
         
         
         
         
         
       5.2.4. Quản lý nhà hàng.** ****

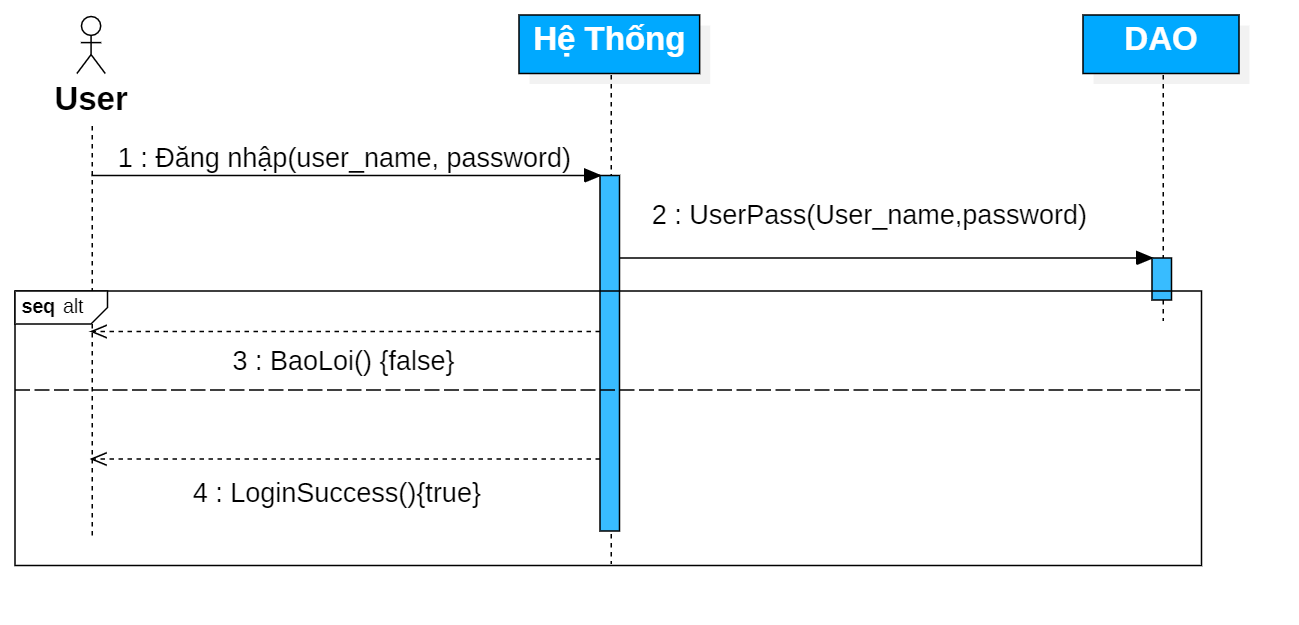




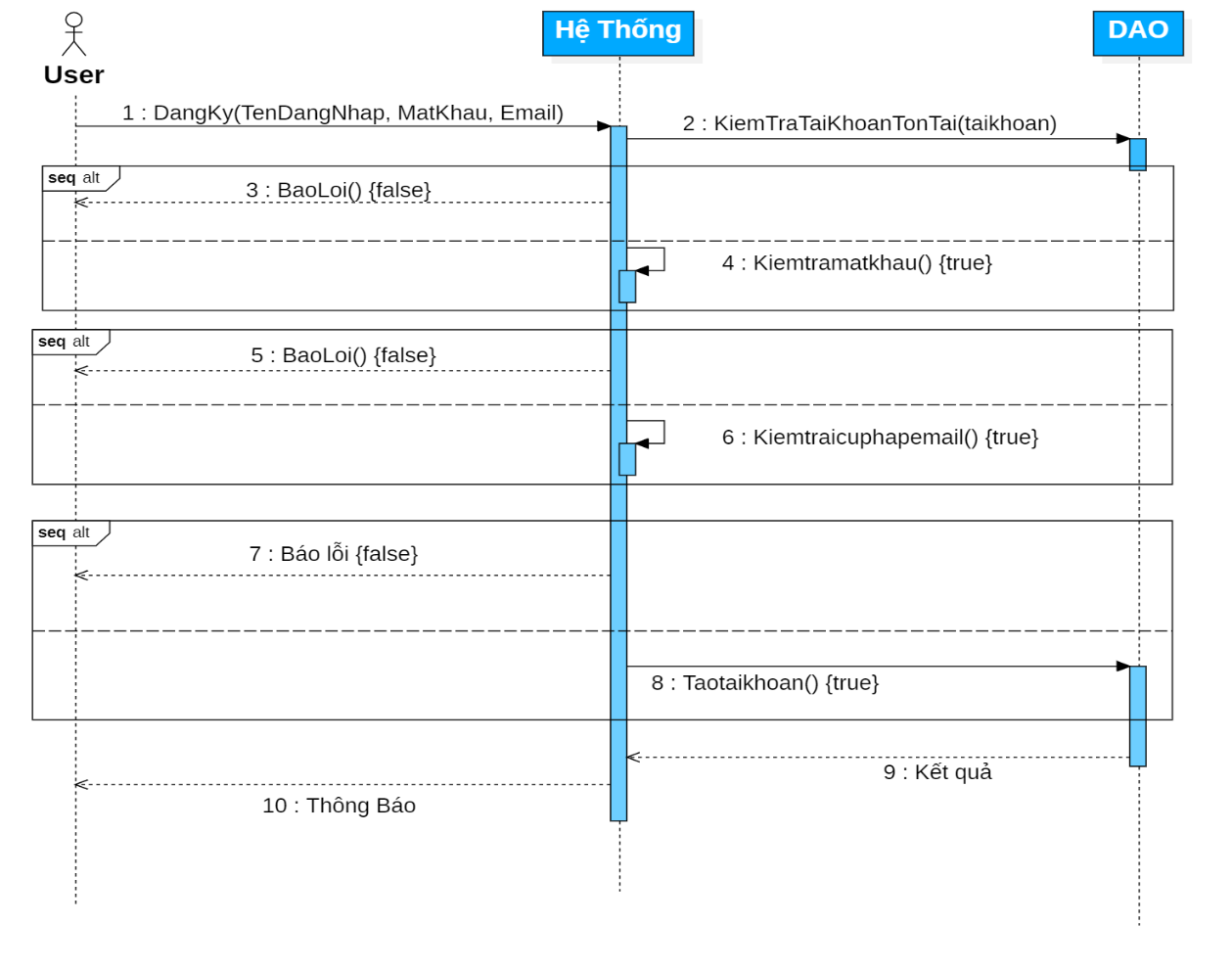
* + 1. **Thông tin tài khoản.**

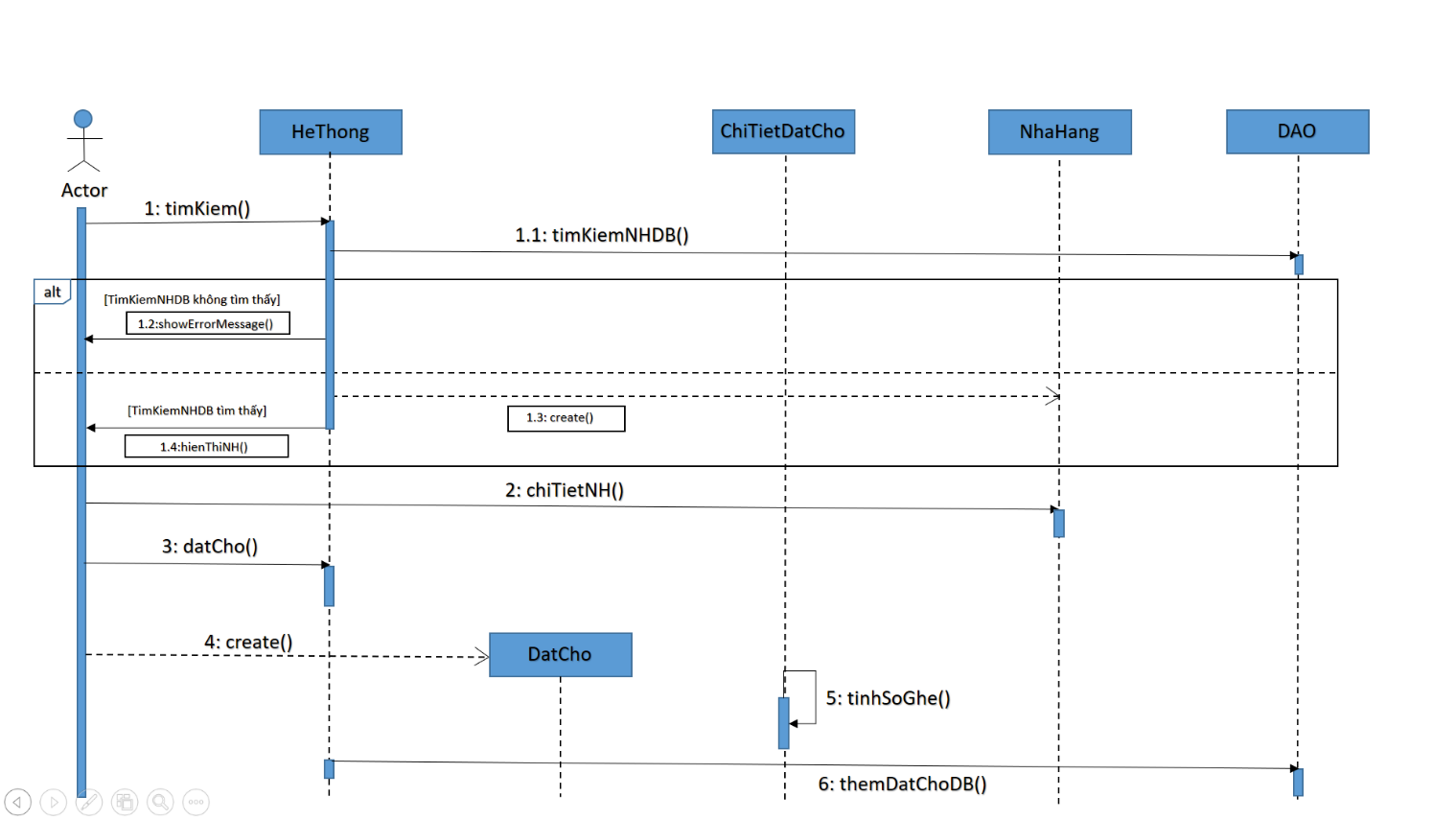


* + 1. **Quản lý tin tức.**
    2. **Quản l** **ý review, liên hệ.**

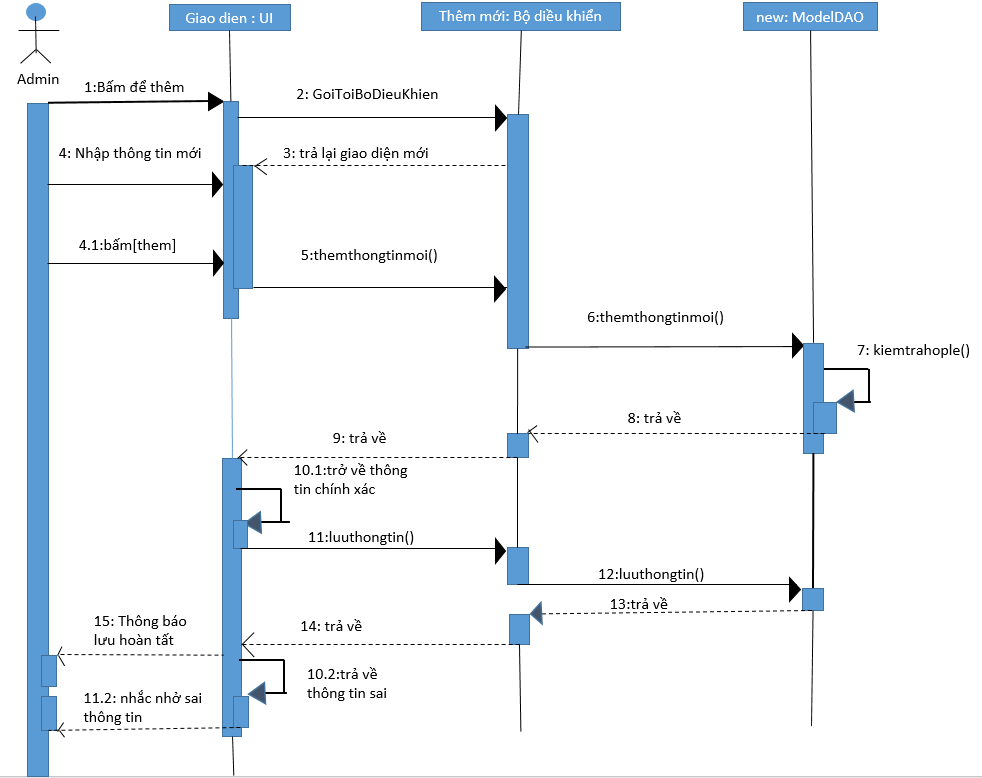
1. **Domain model.**
2. **Sequence diagrams.  
   7.1. Quá trình đăng nhập.**

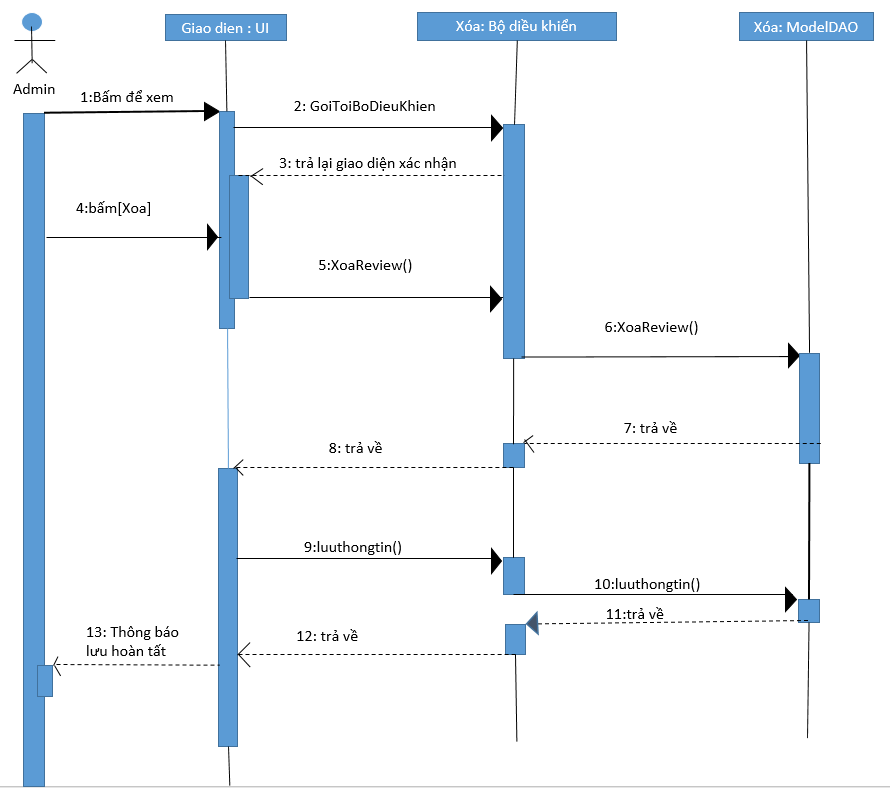
**7.2. Quá trình đăng ký.**



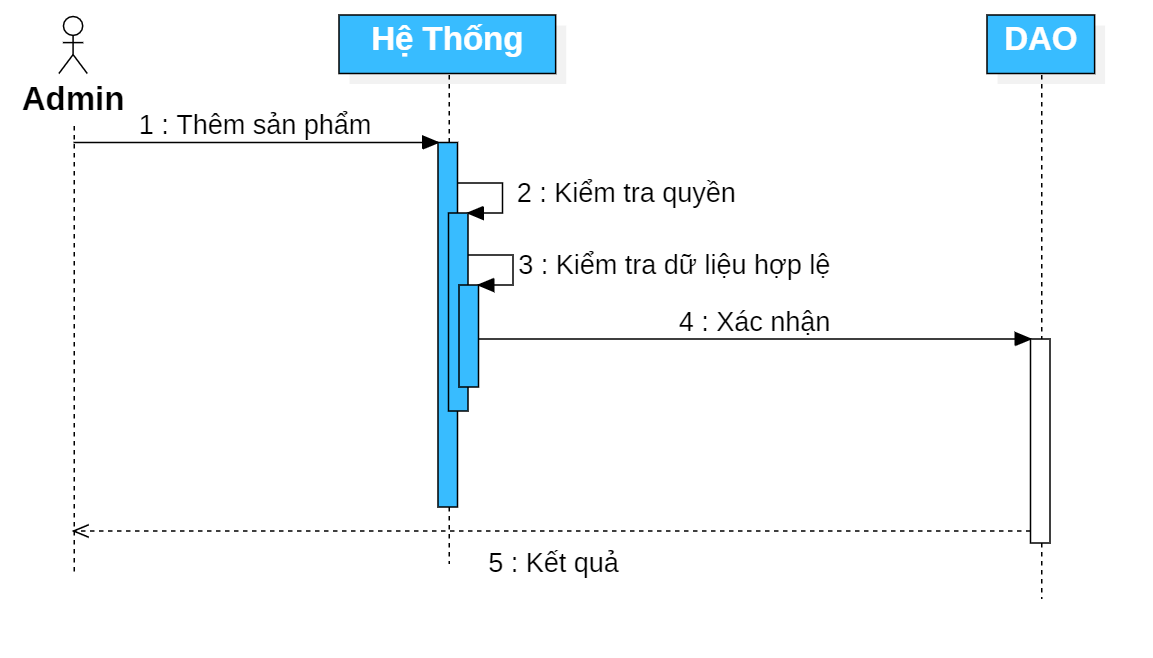


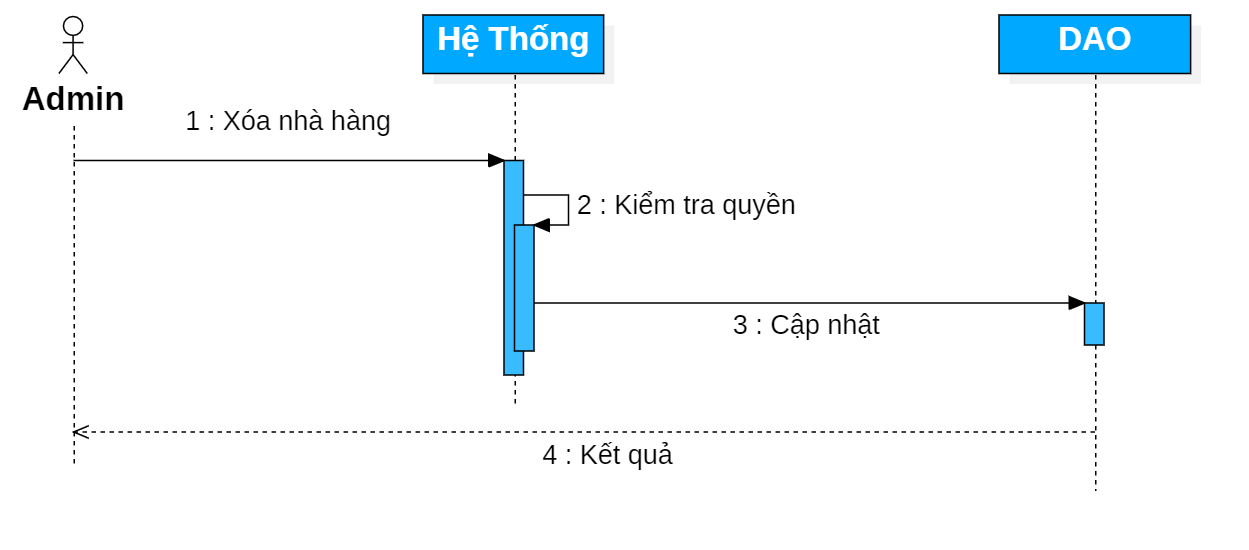
**7.3. Quá trình đặt chỗ**

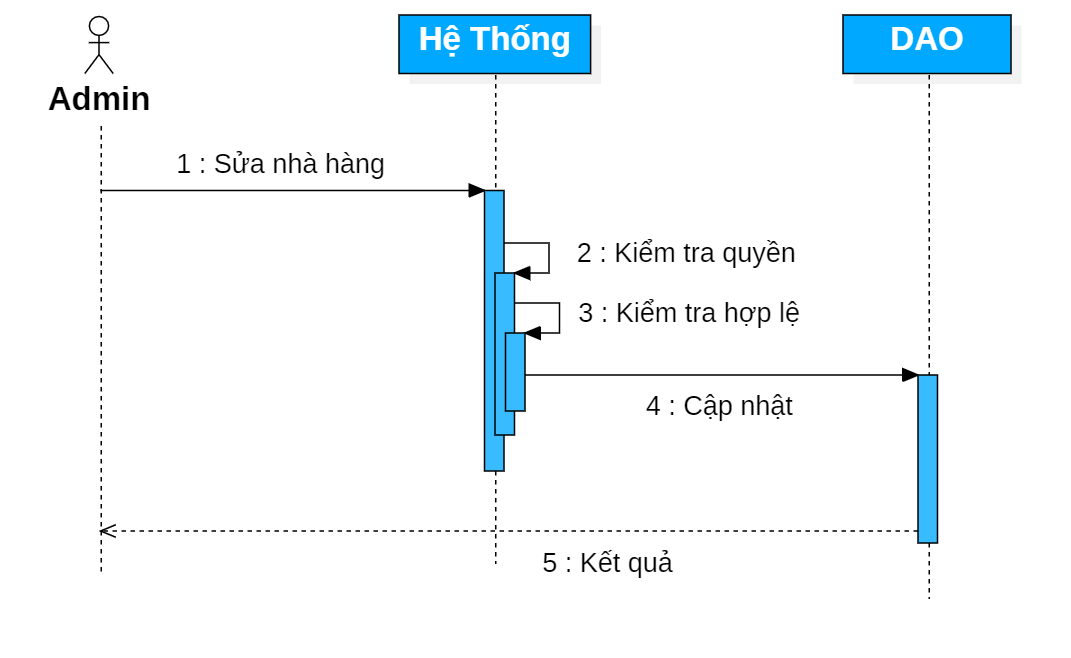
**7.4. Thêm tin tức.**

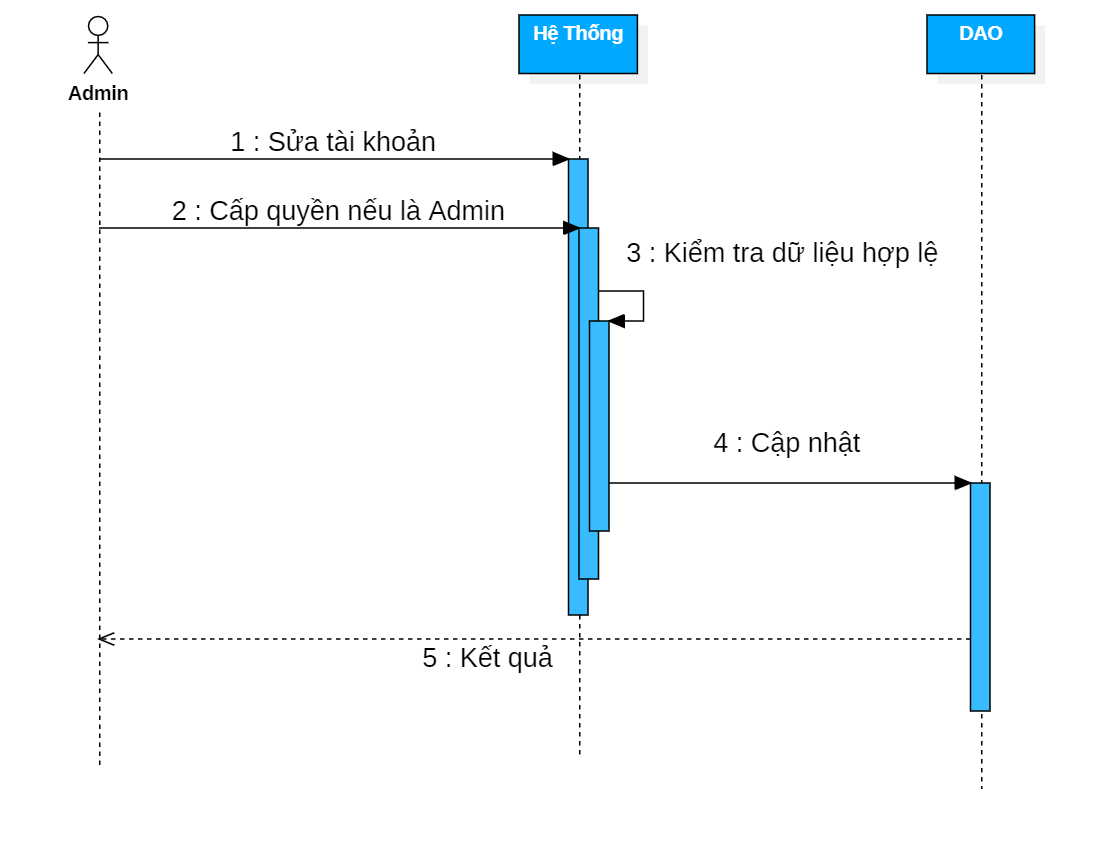
**7.5.Quản lý review.**

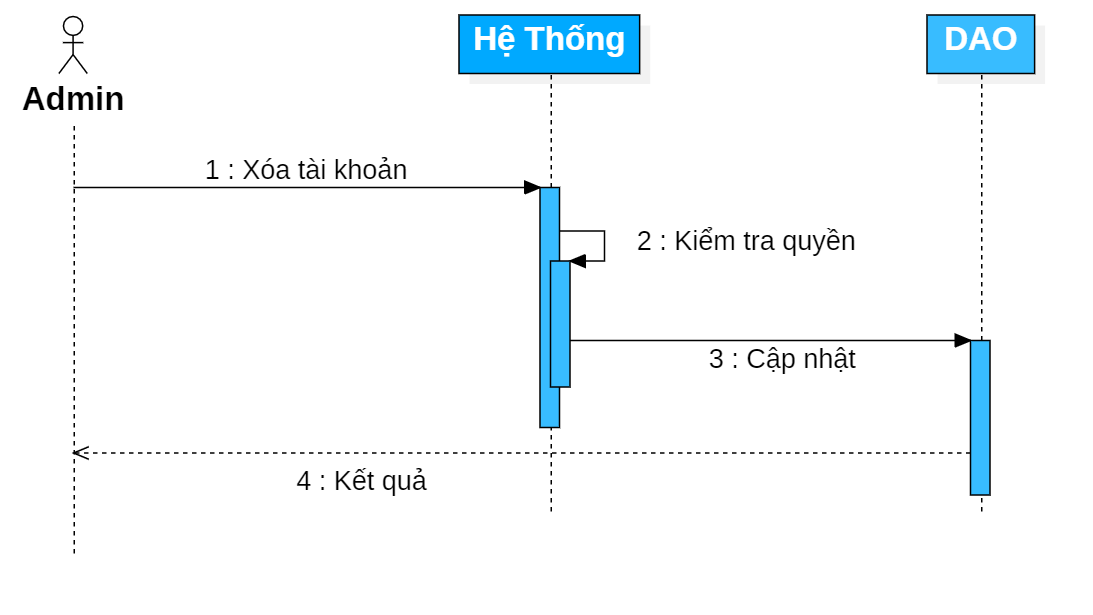
**7.6.Quản lý nhà hàng**

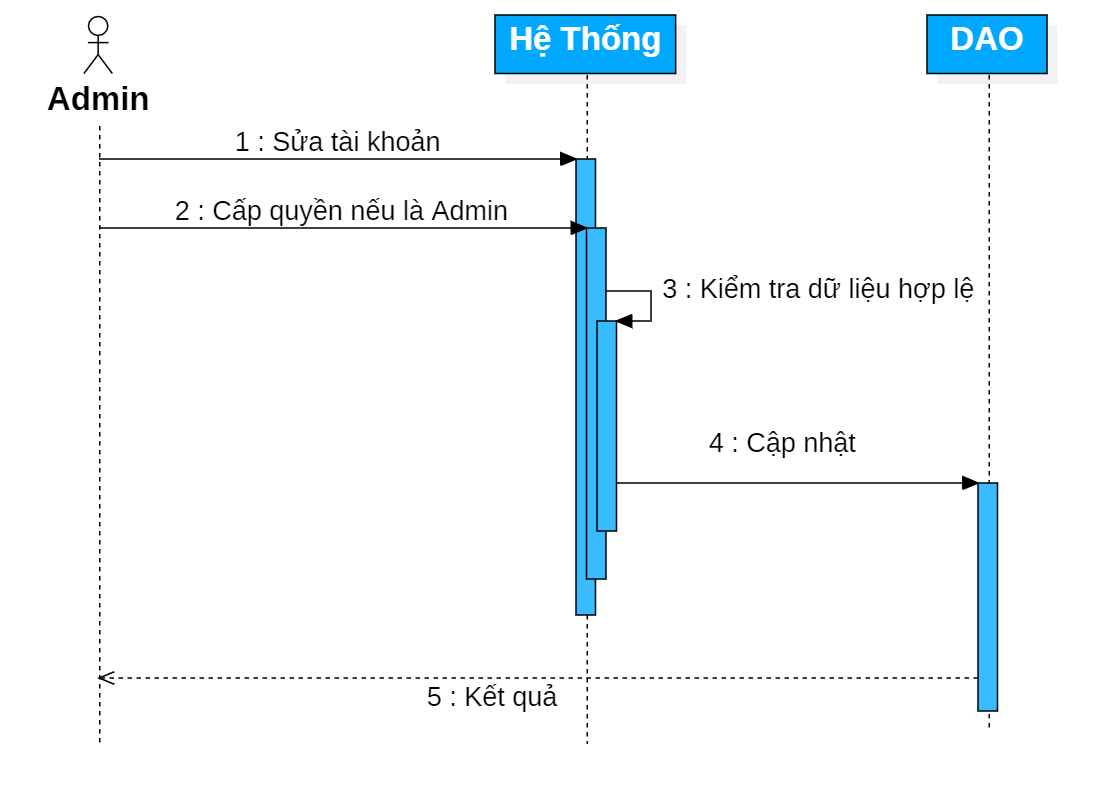




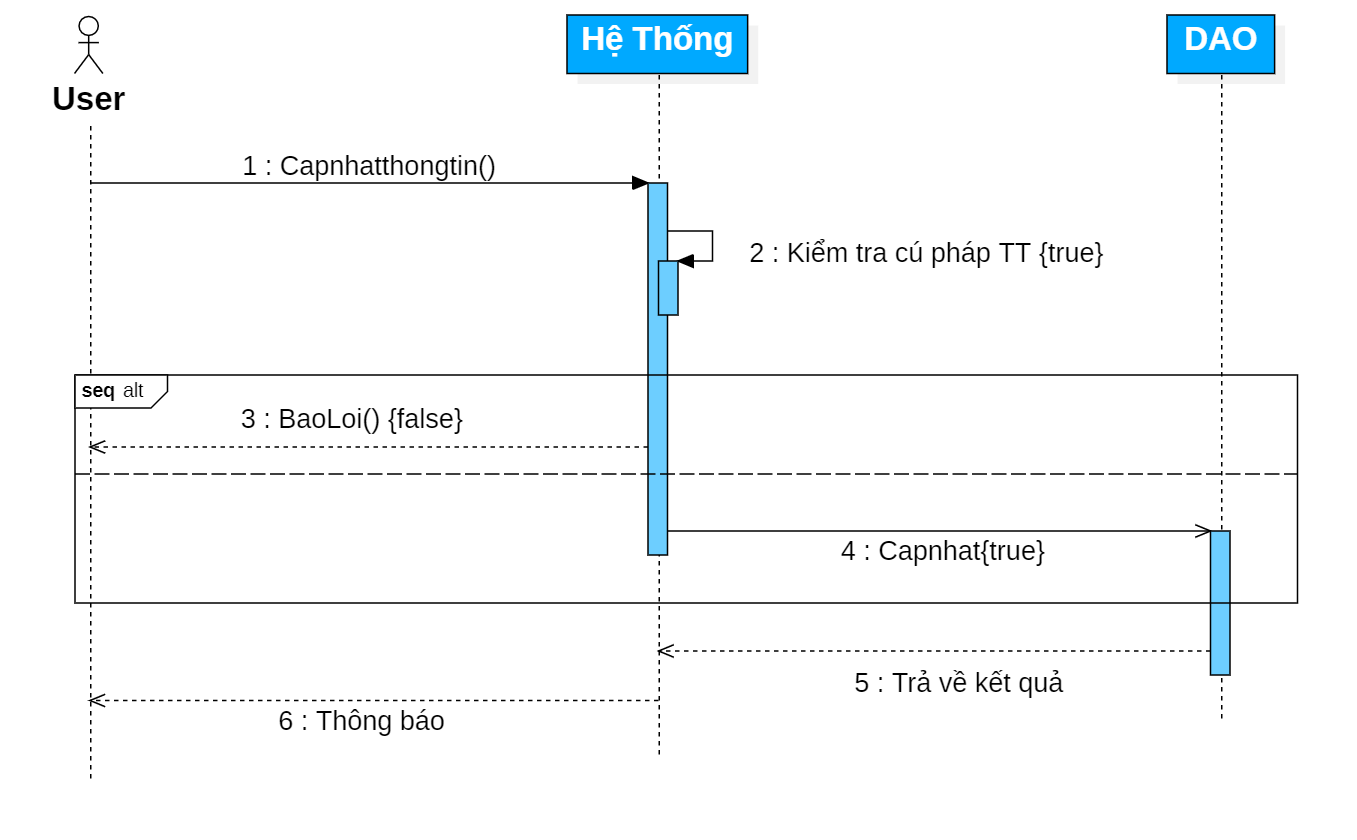


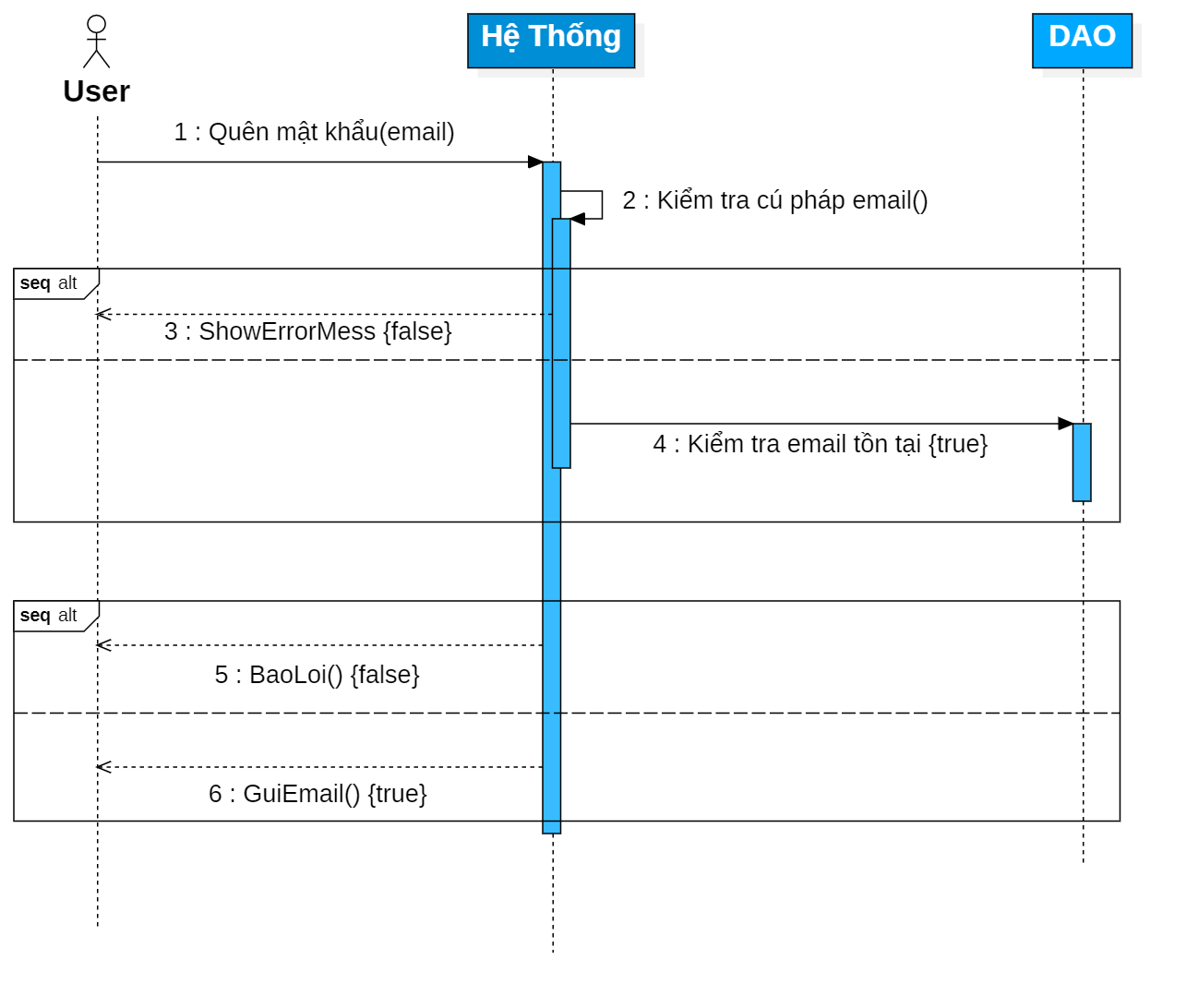
**7.7. Quản lý user**



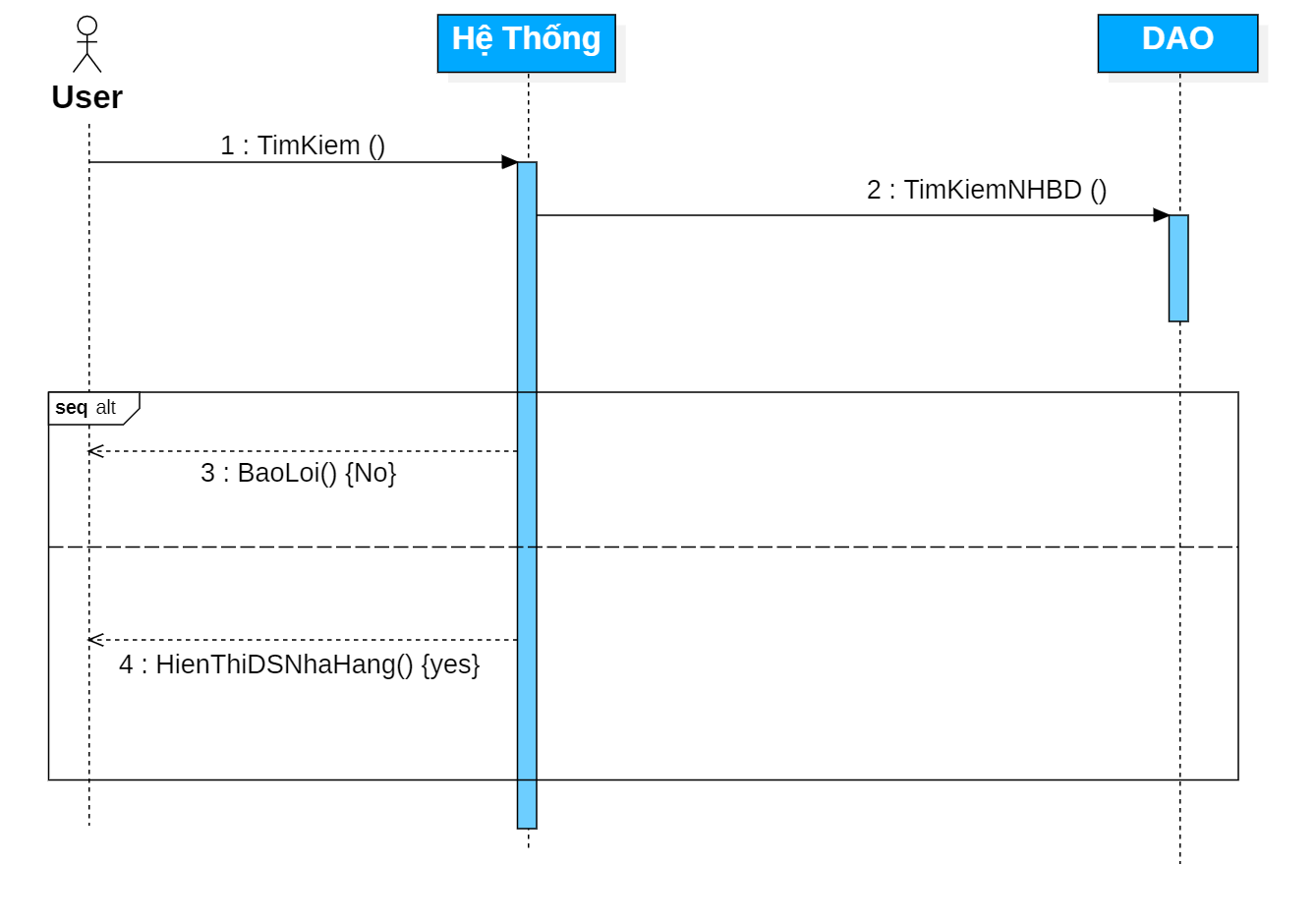


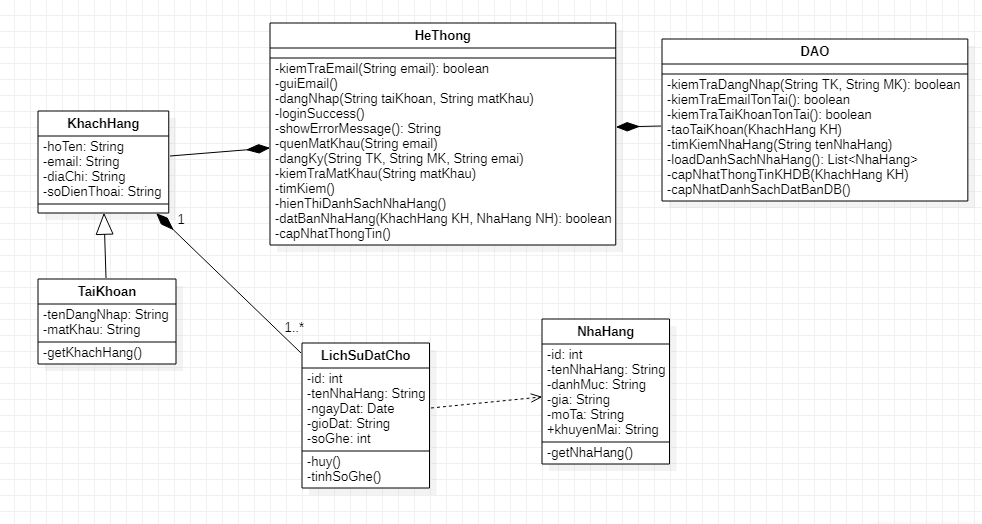
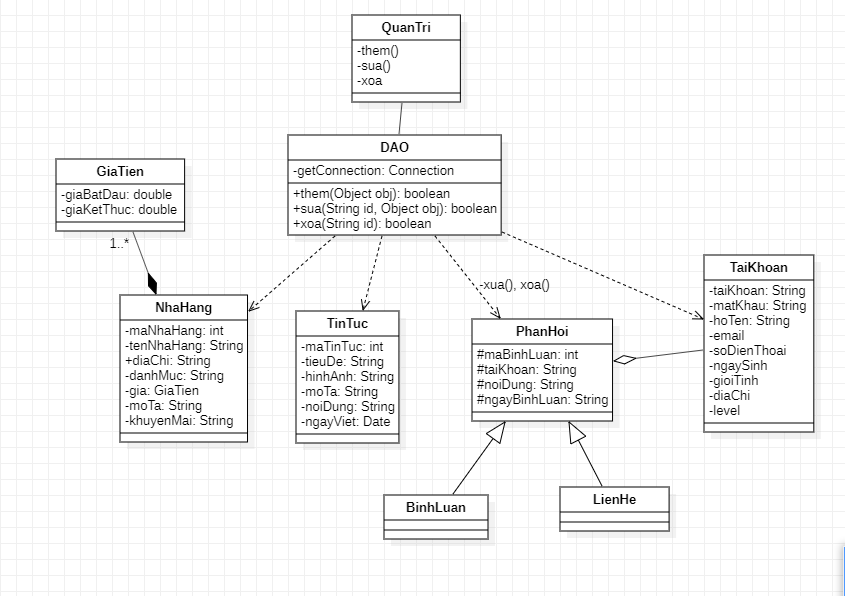
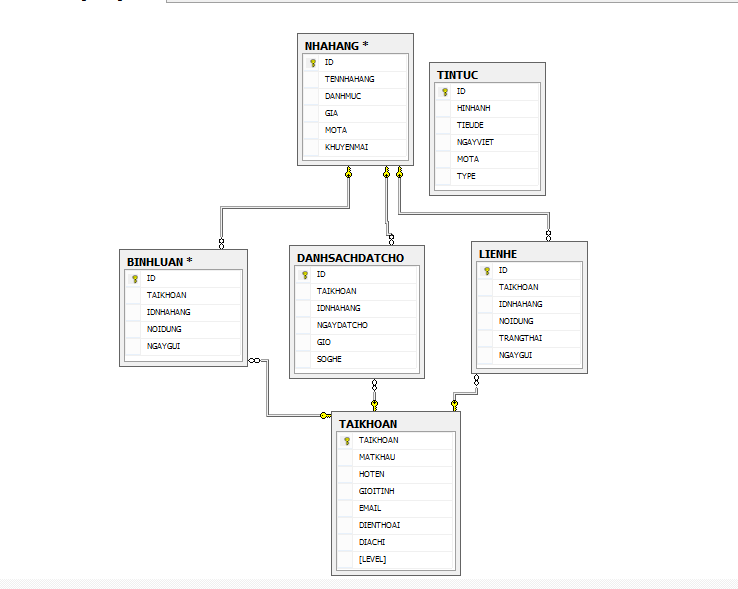
**7.8.Thông tin tài khoản**

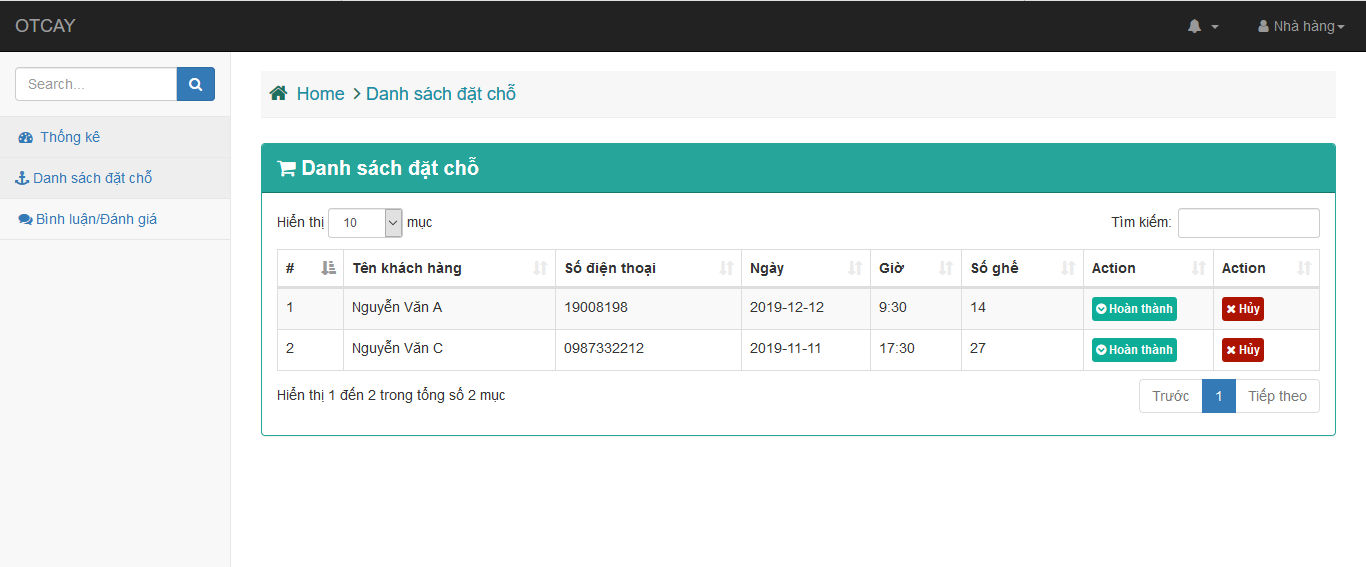
****



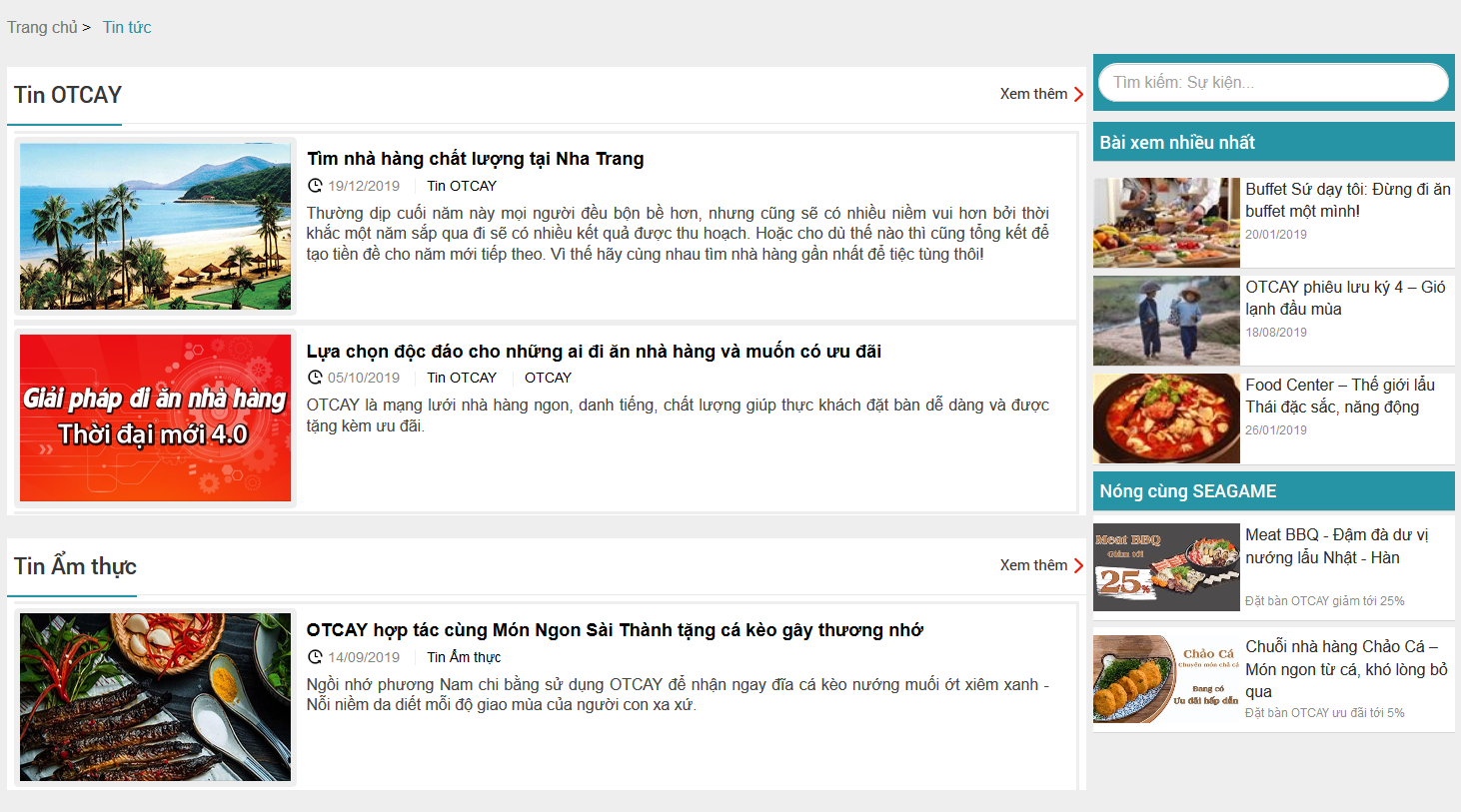
**7.9.Sequence Tìm Kiếm**

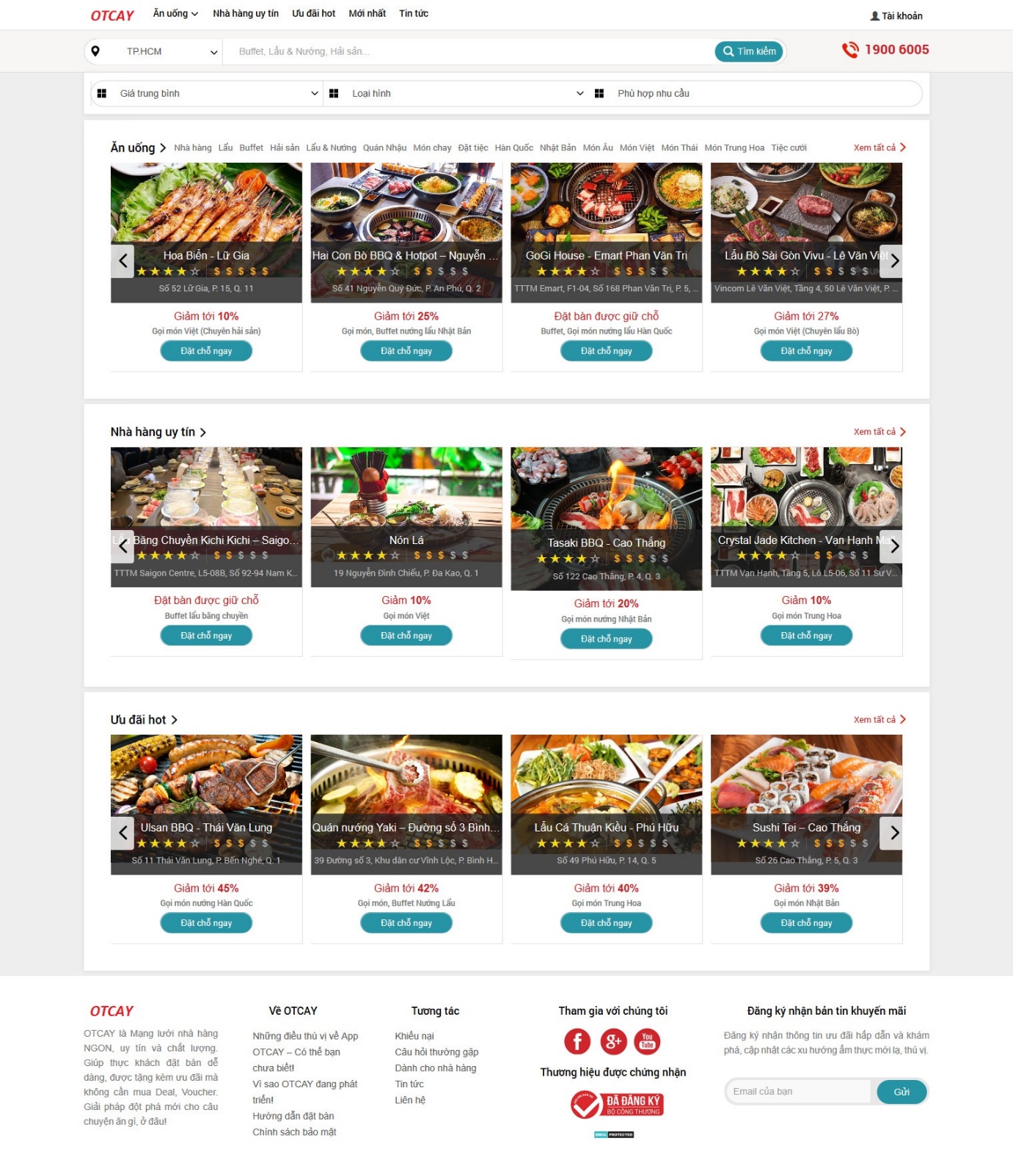
****

1. **Class diagram.**
2. **Database design.**  
   
3. **Thiết kế giao diện người dùng.  
   10.1. Mục tiêu:**- Người dùng cảm thấy dễ nhìn, dễ sử dụng.  
   - Dễ dàng so sánh và chọn lựa giữa những nhà hàng cho phù hợp với nhu cầu.  
   - Dễ dàng sử dụng ngay cả lần đầu.  
   - Chủ nhà hàng quản lý đặt chỗ tối ưu, hình ảnh nhà hàng được thể hiện hấp dẫn. **10.2. Nguyên tắc thiết kế.  
    10.2.1. Nguyên tắc khung giao diện.**

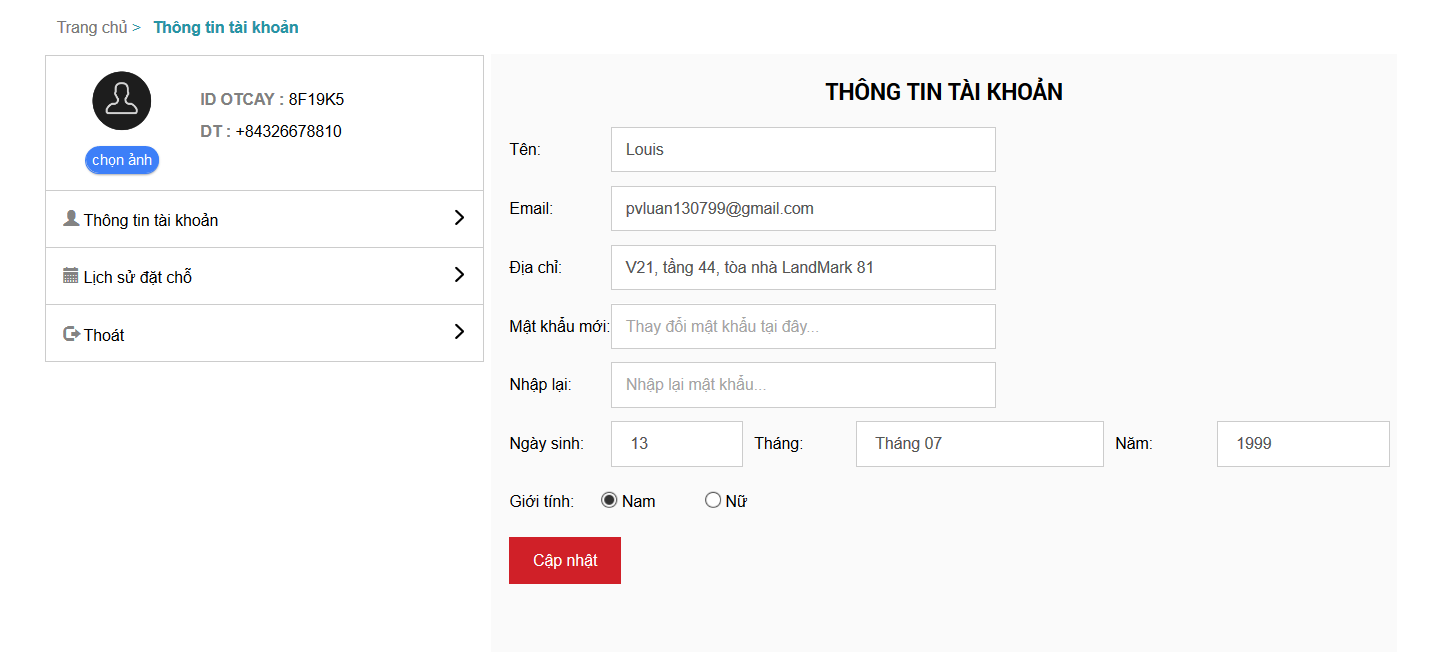


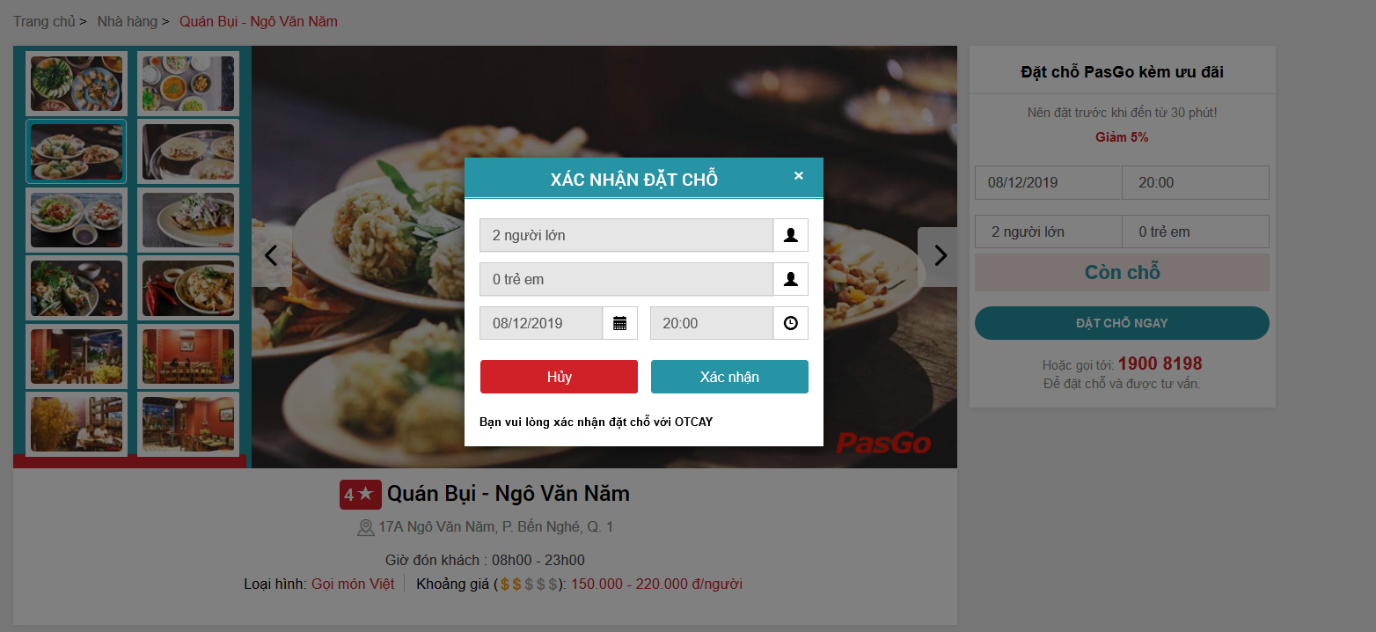
**10.2.2. Nguyên tắc gợi liên quan.**

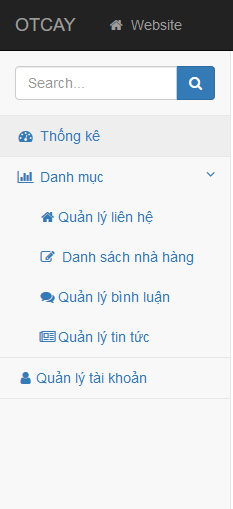


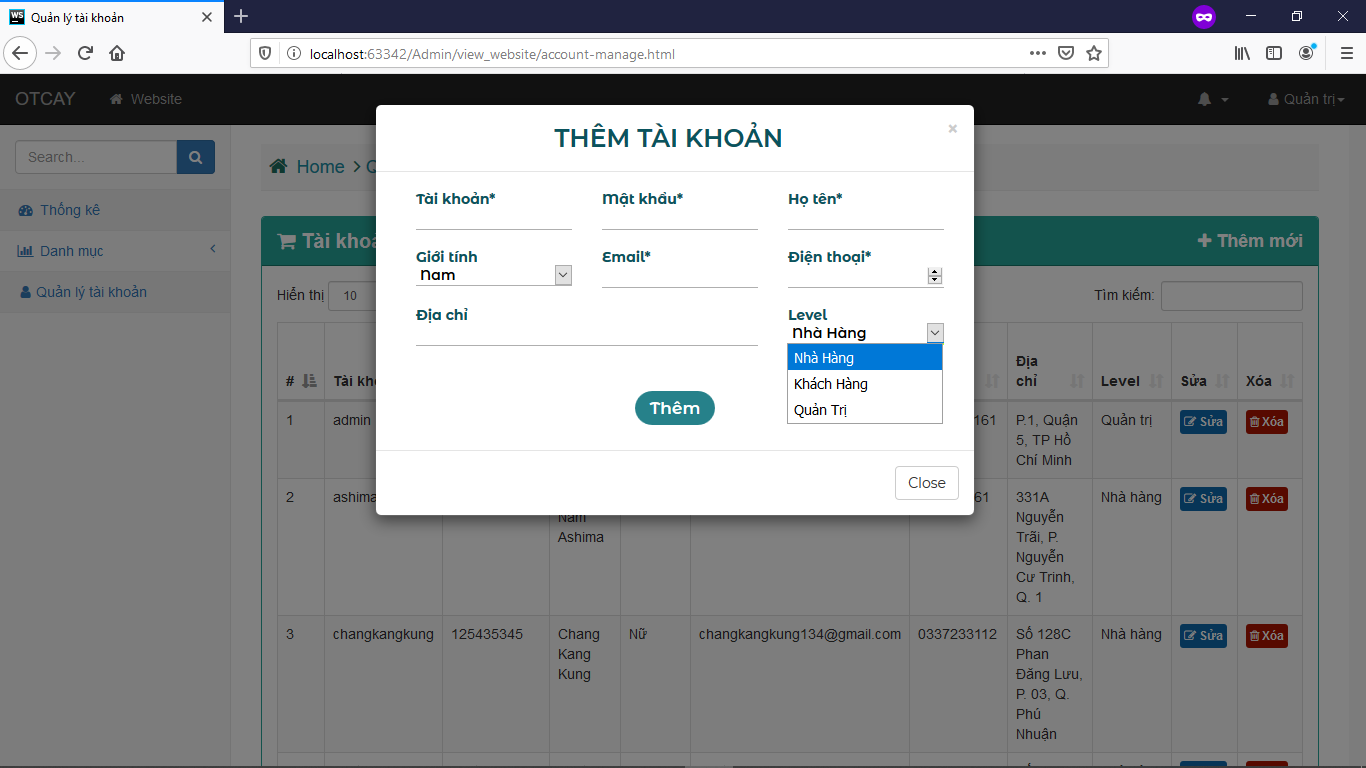
**10.2.3. Nguyên tắc thẩm mỹ.**

**10.2.4. Nguyên tắc logic**

****

* + 1. **Nguyên tắc tiết kiệm thời gian.**





1. **Deloyment diagrams.  
   11.1. Định nghĩa:**

**-Deployment Diagram là bản vẽ giúp chúng ta xác định sẽ triển khai hệ thống phần mềm như thế nào. Đồng thời, xác định chúng ta sẽ đặt các thành phần phần mềm (component) lên hệ thống ra sao.**

-Deployment Diagram thể hiện rõ kiến trúc triển khai nên nó sẽ ảnh hưởng đến sự thiết kế, phát triển, hiệu năng, khả năng mở rộng của hệ thống v.v…

**11.2. Ứng dụng:**Deployment Diagram có thể ứng dụng vào các trường hợp sau:

         - Làm tài liệu để triển khai hệ thống.

         - Sử dụng trong thiết kế kiến trúc cho hệ thống.

         - Dùng trong giao tiếp với khách hàng, các nhà đầu tư.

--- XIN HẾT ---