**BỘ LAO ĐỘNG THƯƠNG BINH VÀ XÃ HỘI  
TRƯỜNG CAO ĐẲNG BÌNH MÌNH SÀI GÒN**

----o0o----



**KHOA CÔNG NGHỆ THÔNG TIN**

**ĐỒ ÁN TỐT NGHIỆP**

WEB QUẢN LÝ KHÁCH SẠN

GVHD: ThS. Trương Bá Vĩnh

SVTH: Nguyễn Hữu Trọng

MSSV: 221ACNTT01

Lớp: Công nghệ thông tin

TP.HCM, Tháng 06 năm 2024

TP. HCM năm 2023

**Lời cảm ơn:**

Tôi xin cảm ơn chân thành đến với những người đã viết tài liệu đã được tham khảo trong bài báo cáo này cũng như giảng viên hướng dẫn của tôi, Nguyễn Quốc Ninh, vì sự chỉ dẫn.

**Cam kết:**

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi và được sự hướng dẫn khoa học của giảng viên Trương Bá Vĩnh. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các Bảng phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong báo cáo còn sử dụng một số khái niệm có trích dẫn và chú thích nguồn gốc. Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung báo cáo của mình.

**NHẬN XÉT GIÁO VIÊN HƯỚNG DẪN**

1. **Họ và tên sinh viên:**
2. **Tên đề tài**:

1. **Nhận xét**:

***Những kết quả đạt được:***

***Những hạn chế:***

1. **Điểm đánh giá** *(theo thang điểm 10, làm tròn đến 0.5):*

Sinh viên:……………………………………………………………………

Điểm số: ……….…… Điểm chữ: …………………………………………

|  |  |
| --- | --- |
|  | *TP. HCM, ngày … tháng … năm 20…*  **Giảng viên chấm thi**  *(Ký và ghi rõ họ tên)* |

# DANH MỤC BẢNG

[Bảng 1: Glossary 7](#_Toc172250293)

[Bảng 2: Use case đăng ký 9](#_Toc172250294)

[Bảng 3 : Use case đăng ký khách sạn 10](#_Toc172250295)

[Bảng 4: Use case đăng nhập 11](#_Toc172250296)

[Bảng 5: Use case tìm khách sạn 12](#_Toc172250297)

[Bảng 6: Use case danh sách các khách sạn 13](#_Toc172250298)

[Bảng 7: Use case trang chủ khách sạn 14](#_Toc172250299)

[Bảng 8: Use case đặt phòng 14](#_Toc172250300)

[Bảng 9: Use case danh sách các phòng 15](#_Toc172250301)

[Bảng 10: Use case thông tin cá nhân 16](#_Toc172250302)

[Bảng 11: Use case xem đánh giá 17](#_Toc172250303)

[Bảng 12: Use case đánh giá 18](#_Toc172250304)

[Bảng 13: Use case thêm phòng 18](#_Toc172250305)

[Bảng 14: Use case chỉnh sửa phòng 19](#_Toc172250306)

[Bảng 15: Use case xóa phòng 20](#_Toc172250307)

[Bảng 16: Use case xem danh sách đặt phòng 21](#_Toc172250308)

[Bảng 17: Use case chỉnh sửa thông tin phòng 22](#_Toc172250309)

[Bảng 18: Use case xóa đặt phòng 23](#_Toc172250310)

[Bảng 19: Use case chỉnh sửa phòng 23](#_Toc172250311)

# DANH MỤC HÌNH

[Hình 1: Use case diagram 8](#_Toc172250275)

[Hình 2: Sequence diagram đăng ký 30](#_Toc172250276)

[Hình 3: Sequence diagram đăng ký khách sạn 30](#_Toc172250277)

[Hình 4: Sequence diagram đăng nhập 31](#_Toc172250278)

[Hình 5: Sequence diagram tìm khách sạn 31](#_Toc172250279)

[Hình 6: Sequence diagram danh sách khách sạn 31](#_Toc172250280)

[Hình 7: Sequence diagram trang chủ khách sạn 31](#_Toc172250281)

[Hình 8: Sequence diagram đặt phòng 31](#_Toc172250282)

[Hình 9: Sequence diagram danh sách các phòng 32](#_Toc172250283)

[Hình 10: Sequence diagram thông tin cá nhân 32](#_Toc172250284)

[Hình 11: Sequence diagram xem đánh giá 32](#_Toc172250285)

[Hình 12: Lược đồ thực thể kết gợp(ERD) 34](#_Toc172250286)

[Hình 13: Lược đồ CSDL / DataModel 35](#_Toc172250287)

MỤC LỤC

[DANH MỤC BẢNG 4](#_Toc172250312)

[DANH MỤC HÌNH 5](#_Toc172250313)

[MỞ ĐẦU 1](#_Toc172250314)

[1. Giới thiệu sơ lược về đề tài 1](#_Toc172250315)

[2. Mục đích và ý nghĩa 1](#_Toc172250316)

[2.1. Mục đích: 1](#_Toc172250317)

[2.2. Ý nghĩa: 1](#_Toc172250318)

[3. Lý do chọn đề tài: 2](#_Toc172250319)

[4. Mục tiêu và nhiệm vụ 2](#_Toc172250320)

[5. Phạm vi đề tài 3](#_Toc172250321)

[6. Cấu trúc đề tài 3](#_Toc172250322)

[NỘI DUNG 4](#_Toc172250323)

[CHƯƠNG 1: REQUIREMENTS 4](#_Toc172250324)

[1.1. Functional requirements 4](#_Toc172250325)

[1.1.1. Đăng ký 4](#_Toc172250326)

[1.1.2. Đăng ký khách sạn 4](#_Toc172250327)

[1.1.3 Đăng nhập 4](#_Toc172250328)

[1.1.4. Tìm khách sạn 4](#_Toc172250329)

[1.1.5. Danh sách các khách sạn 4](#_Toc172250330)

[1.1.6. Trang chủ của khách sạn 4](#_Toc172250331)

[1.1.7. Đặt phòng 4](#_Toc172250332)

[1.1.8. Danh sách các phòng 5](#_Toc172250333)

[1.1.9. Thông tin cá nhân 5](#_Toc172250334)

[1.1.10. Xem đánh giá 5](#_Toc172250335)

[1.1.11. Đánh giá 5](#_Toc172250336)

[1.1.12. Thêm phòng 5](#_Toc172250337)

[1.1.13.Chỉnh sửa phòng 5](#_Toc172250338)

[1.1.14. Xóa phòng 5](#_Toc172250339)

[1.1.15. Xem danh sách những cuộc đặt phòng 5](#_Toc172250340)

[1.1.16. Chỉnh sửa thông tin đặt phòng 6](#_Toc172250341)

[1.1.17. Xóa đặt phòng 6](#_Toc172250342)

[1.1.18. Chỉnh sửa thông tin khách sạn 6](#_Toc172250343)

[1.2. Non-Funtional requirements 6](#_Toc172250344)

[1.2.1. Tính sẵn có: 6](#_Toc172250345)

[1.2.2. Tính dễ dùng: 6](#_Toc172250346)

[1.2.3. Tính ổn định: 6](#_Toc172250347)

[1.2.4. Hiệu suất: 6](#_Toc172250348)

[1.2.5. Sự hỗ trợ: 7](#_Toc172250349)

[1.2.6. Các rằng buộc thiết kế: 7](#_Toc172250350)

[1.3. Glossary 7](#_Toc172250351)

[CHƯƠNG 2: USE CASE MODEL 8](#_Toc172250352)

[2.1. Use case diagram 8](#_Toc172250353)

[2.2. Use case specification / scemarios 9](#_Toc172250354)

[2.2.1. Use case Đăng ký 9](#_Toc172250355)

[2.2.2. Use case đăng ký khách sạn 9](#_Toc172250356)

[2.2.3. Use case đăng nhập 10](#_Toc172250357)

[2.2.4. Use case tìm khách sạn 11](#_Toc172250358)

[2.2.5. Use case danh sách các khách sạn 12](#_Toc172250359)

[2.2.6. Use case trang chủ khách sạn 13](#_Toc172250360)

[2.2.7. Use case đặt phòng 14](#_Toc172250361)

[2.2.8. Use case danh sách các phòng 15](#_Toc172250362)

[2.2.9. Use case thông tin các nhân 15](#_Toc172250363)

[2.2.10. Use case xem đánh giá 16](#_Toc172250364)

[2.2.11. Use case đánh giá 17](#_Toc172250365)

[2.2.12. Use case thêm phòng 18](#_Toc172250366)

[2.2.13. Use case chỉnh sửa phòng 19](#_Toc172250367)

[2.2.14. Use case xóa phòng 20](#_Toc172250368)

[2.2.15. Use case xem danh sách đặt phòng 20](#_Toc172250369)

[2.2.16. Use case chỉnh sửa thông tin đặt phòng 21](#_Toc172250370)

[2.2.17. Use case xóa đặt phòng 22](#_Toc172250371)

[2.2.18. Use case chỉnh sửa thông tin khách sạn 23](#_Toc172250372)

[CHƯƠNG 3: SYSTEM ARCHITECT 24](#_Toc172250373)

[3.1.Phatform 24](#_Toc172250374)

[3.2.Lựa chọn ngôn ngữ 24](#_Toc172250375)

[3.3.Lựa chọn framework và thư viện 24](#_Toc172250376)

[3.3.1. Giới thiệu về Reactjs 24](#_Toc172250377)

[3.3.2. Giới thiệu về Spring boot 25](#_Toc172250378)

[3.3.3. Giới thiệu Spring Security 26](#_Toc172250379)

[3.3.4. Giới thiệu Hibernate 27](#_Toc172250380)

[3.4.Lựa chọn Database 28](#_Toc172250381)

[3.5.Lựa chọn Deloyment 28](#_Toc172250382)

[3.6.Lựa chọn Monitoring 29](#_Toc172250383)

[CHƯƠNG 4: SYSTEM BEHAVIOR 30](#_Toc172250384)

[4.1. Sequence diagram đăng ký 30](#_Toc172250385)

[4.2. Sequence diagram Đăng ký khách sạn 30](#_Toc172250386)

[4.3. Sequence diagram đăng nhập 31](#_Toc172250387)

[4.4. Sequence diagram tìm khách sạn 31](#_Toc172250388)

[4.5. Sequence diagram danh sách khách sạn 31](#_Toc172250389)

[4.6. Sequence diagram trang chủ khách sạn 31](#_Toc172250390)

[4.7. Sequence diagram đặt phòng 31](#_Toc172250391)

[4.8. Sequence diagram danh sách các phòng 32](#_Toc172250392)

[4.9. Sequence diagram thông tin cá nhân 32](#_Toc172250393)

[4.10. Sequence diagram xem đánh giá 32](#_Toc172250394)

[4.11. Sequence diagram đánh giá 32](#_Toc172250395)

[4.12. Sequence diagram thêm phòng 32](#_Toc172250396)

[4.13. Sequence diagram chỉnh sửa phòng 32](#_Toc172250397)

[4.14. Sequence diagram xóa phòng 32](#_Toc172250398)

[4.15. Sequence diagram xem danh sách những phòng đã đặt 32](#_Toc172250399)

[4.16. Sequence diagram chỉnh sửa thông tin đặt phòng 33](#_Toc172250400)

[4.17. Sequence diagram xóa đặt phòng 33](#_Toc172250401)

[4.18. Sequence diagram chỉnh sửa thông tin khách sạn 33](#_Toc172250402)

[CHƯƠNG 5: SYSTEM DATA MODEL 34](#_Toc172250403)

[5.1. Lược đồ thực thể kết gợp(ERD): 34](#_Toc172250404)

[5.2. Lược đồ CSDL / DataModel 35](#_Toc172250405)

[TỔNG KẾT 36](#_Toc172250406)

# MỞ ĐẦU

## 1. Giới thiệu sơ lược về đề tài

Hiện nay, ngành khách sạn là một trong những ngành công nghiệp lớn và liên tục phát triển, đòi hỏi sự quản lý chuyên nghiệp cũng như quảng bá hiệu quả. Trong bối cảnh cạnh tranh ngày càng gay gắt, việc áp dụng công nghệ vào quản lý và vận hành khách sạn không chỉ giúp nâng cao hiệu quả kinh doanh mà còn đáp ứng tốt hơn nhu cầu của khách hàng. Một trang web quản lý khách sạn, với tính năng đa dạng và tiện lợi, chính là một công cụ vô cùng cần thiết để gia tăng hiệu quả quản lý cũng như nâng cao trải nghiệm của cả người dùng và người quản lý.

## 2. Mục đích và ý nghĩa

### 2.1. Mục đích:

Xây dựng Website phục vụ cho nhu cầu đặt phòng khách sạn cũng như là quản bá nó. Người đặt phòng chỉ cần tốn ít thời gian để có thể tìm một căn phòng ưng ý ở nơi mà mình muốn, và họ cũng có thể xem các đánh giá của người khác về phòng đó. Những chủ khách sạn có thể đăng phòng của mình lên trên Website để quản bá và quản lý. Ngoài những chức năng tiện lợi ra thì Website cũng có hệ thống bảo mật an toàn cho khách hàng cũng như là nhưng chủ quản lý khách sạn.

### 2.2. Ý nghĩa:

Đề tài xây dựng và thiết kế trang web quản lý khách sạn mang ý nghĩa to lớn không chỉ đối với việc học tập và nghiên cứu mà còn đối với ngành công nghiệp khách sạn nói chung.

Website quản lý khách sạn giúp nâng cao hiệu quả hoạt động kinh doanh và quản lý trong các khách sạn, việc áp dụng công nghệ vào quản lý khách sạn không chỉ giúp tiết kiệm chi phí và thời gian mà còn tăng cường hiệu quả cạnh tranh của các khách sạn trên thị trường

Tạo trải nghiệm tốt cho khách hàng với tính năng tìm và đặt phòng trực tuyến, khách hàng có thể dễ dàng tìm và đặt phong theo nhu cầu của mình một cách nhanh chóng và tiện lợi. Họ có thể xem trước thông tin một cách chi tiết và những đánh giá từ những khách hàng trước đó, giúp họ có quyết định đúng hơn.

Hệ thống trang Web còn được thiết kế với các biện pháp bảo mật, bảo vệ thông tin các nhân và dữ liệu giao dịch của khách hàng cũng như chủ khách sạn.

## 3. Lý do chọn đề tài:

Hiện nay, ngành khách sạn là một trong những ngành công nghiệp lớn và liên tục phát triển, đòi hỏi sự quản lý chuyên nghiệp cũng như quảng bá hiệu quả. Trong bối cảnh cạnh tranh ngày càng gay gắt, việc áp dụng công nghệ vào quản lý và vận hành khách sạn không chỉ giúp nâng cao hiệu quả kinh doanh mà còn đáp ứng tốt hơn nhu cầu của khách hàng. Một trang web quản lý khách sạn, với tính năng đa dạng và tiện lợi, chính là một công cụ vô cùng cần thiết để gia tăng hiệu quả quản lý cũng như nâng cao trải nghiệm của cả người dùng và người quản lý.

## 4. Mục tiêu và nhiệm vụ

**Mục tiêu:**

* + Nắm bắt được các yêu cầu cụ thể của khách hàng cũng như của những chủ khách sạn mà đưa ra những thiết kế hệ thống phù hợp.
  + Lên kế hoạch và phân bổ công việc một cách có hệ thống.
  + Hoàn thiện dự án trong thời gian 6 tuần.

**Nhiệm vụ:**

* + Tiến hành khảo sát nhu cầu hiện nay của những đối tượng người dùng để liên kế hoạch thiết kế Database, phác thảo UI.
  + Lựa ngôn ngữ lập trình và Framework dựa vào những ý tưởng đang có.
  + Thống nhất ngôn ngữ lập trình, Framework và các ý tưởng để tiến hành xây dựng nền móng cơ bản cho Website.
  + Viết Website và xây dựng những tính năng cốt lõi.
  + Chỉnh sửa, nâng cấp và vá lỗi.
  + Thử nghiệm và hoàn thiện Website.

## 5. Phạm vi đề tài

Vì độ rộng của đề tài thế nên phạm vi đề tài sẽ dừng lại ở quy mô nhỏ và bị hạn chế bởi khối lượng kiến thức cũng như thời lượng chương trình.

## 6. Cấu trúc đề tài

Đề tài được chia ra thành 6 chương:

CHƯƠNG 1: REQUIREMENTS

CHƯƠNG 2: USE-CASE MODEL

CHƯƠNG 3: SYSTEM ARCHITECT

CHƯƠNG 4: SYSTEM BEHAVIOR

CHƯƠNG 5: SYSTEM DATA MODEL

CHƯƠNG 6: SYSTEM USER INTERFACES

# NỘI DUNG

# CHƯƠNG 1: REQUIREMENTS

## 1.1. Functional requirements

### 1.1.1. Đăng ký

Mô tả: Khi có một người dùng muốn đăng ký một tài khoản để tiện cho việc đặt phòng và theo dõi lịch sử các phòng đã được đặt.

Tài khoản: mỗi tài khoản sẽ có mã Id, họ và tên đệm, tên, email, mật khẩu, số điện thoại.

### 1.1.2. Đăng ký khách sạn

Mô tả: khi có người muốn đăng khách sạn của mình lên trang web.

Khách sạn: mỗi khách sạn sẽ có mã Id, tên khách sạn, thành phố, địa chỉ, mô tả cả khách sạn, số điện thoại, ảnh của khách sạn đó.

### 1.1.3 Đăng nhập

Mô tả: Để người dùng truy cập vào tài khoản của mình.

### 1.1.4. Tìm khách sạn

Mô tả: Giúp gia tăng trải nghiệm người dùng bằng cách tìm khách sạn dựa trên thành phố và ngày đặt.

### 1.1.5. Danh sách các khách sạn

Mô tả: Hiển thị tất cả các phòng dựa trên request được gửi đến hệ thống.

### 1.1.6. Trang chủ của khách sạn

Mô tả: Đây là nơi để xem toàn bộ thông tin của khách sạn, đánh giá, tất cả các phòng để đặt.

### 1.1.7. Đặt phòng

Mô tả: Khi người dùng muốn đặt một phòng, chức năng nay có thể dùng khi có hoặc không có tài khoản.

Đặt phòng: mỗi phòng được đặt sẽ có Id, ngày đặt phòng, ngày trả phòng, tổng số lượng khách, họ tên đầy đủ của khách, email , mã đặt phòng, số điện thoại.

### 1.1.8. Danh sách các phòng

Mô tả: Đây là một thành phần của trang web sẽ xuất hiện tại trang quản lý các phòng của chủ khách sạn, trang của khách sạn, và trang chủ của web.

Phòng: mỗi phòng sẽ có Id phòng, ảnh, mô tả phòng, giá phòng, loại phòng.

### 1.1.9. Thông tin cá nhân

Mô tả: Để người dùng xem thông tin cá nhân, vai trò của mình lẫn lịch sử phòng.

### 1.1.10. Xem đánh giá

Mô tả: Đây là chức năng để người dùng xem tất cả đánh giá của một khách sạn bất kì.

### 1.1.11. Đánh giá

Mô tả: Khi người dùng muốn đánh giá một khách sạn sau khi đặt phòng.

Đánh giá: mỗi đánh giá sẽ có Id, số sao, ngày đánh giá, nhận xét.

### 1.1.12. Thêm phòng

Mô tả: Khi chủ khách sạn cần thêm một phòng mới vô trong trang khách sạn của mình.

### 1.1.13.Chỉnh sửa phòng

Mô tả: Khi chủ khách sạn muốn chỉnh lại thông tin của phòng cụ thể.

### 1.1.14. Xóa phòng

Mô tả: Khi chủ khách sạn muốn xóa một phòng cụ thể.

### 1.1.15. Xem danh sách những cuộc đặt phòng

Mô tả: Khi chủ khách sạn muốn xem lại toàn bộ danh sách các cuộc đặt phòng trong khách sạn của mình.

### 1.1.16. Chỉnh sửa thông tin đặt phòng

Mô tả: Khi có người dùng chủ khách sạn hoặc chủ khách sạn muốn chỉnh lại thông tin của một cuộc đặt phòng cụ thể.

### 1.1.17. Xóa đặt phòng

Mô tả: Khi chủ khách sạn muốn xóa một cuộc đợt phòng cụ thể.

### 1.1.18. Chỉnh sửa thông tin khách sạn

Mô tả: Khi chủ khách sạn muốn sửa lại thông tin khách sạn của mình.

## 1.2. Non-Funtional requirements

### 1.2.1. Tính sẵn có:

- Hỗ trợ nhiều người dùng đồng thời.

- Lưu toàn bộ thông tin vào hệ thống mỗi khi có người dùng đặt phòng, quản lý khách sạn.

### 1.2.2. Tính dễ dùng:

- Giao diện người dùng tương thích với toàn bộ trình duyệt web.

### 1.2.3. Tính ổn định:

- Hệ thống phải hoạt động liên tục 24 giờ/7 ngày, 7 ngày/ tuần, với thời gian ngừng hoạt động không quá 5%

### 1.2.4. Hiệu suất:

- Hệ thống phải hỗ trợ đến 1000 người dùng truy xuất CSDL trung tâm đồng thời cùng lúc, và đến 500 người dùng truy xuất các server cục bộ.

- Hệ thống phải truy xuất đến CSDL danh mục lịch sử hóa đơn với độ trễ không quá 5 giây.

- Hệ thống phải có khả năng hoàn tất 95% order của thực khách trong vòng 2 phút

- Hệ thống thêm, chỉnh sửa, xóa những dữ liệu mới từ người dùng như nhân viên, nhà hàng,…dưới 5 giây

### 1.2.5. Sự hỗ trợ:

- Không có

### 1.2.6. Các rằng buộc thiết kế:

- Hệ thống tích hợp với hệ thống nhà hàng có sẵn, một CSDL RDBMS.

- Hệ thống phải công cấp giao diện dựa web, hệ điều hành điện thoại.

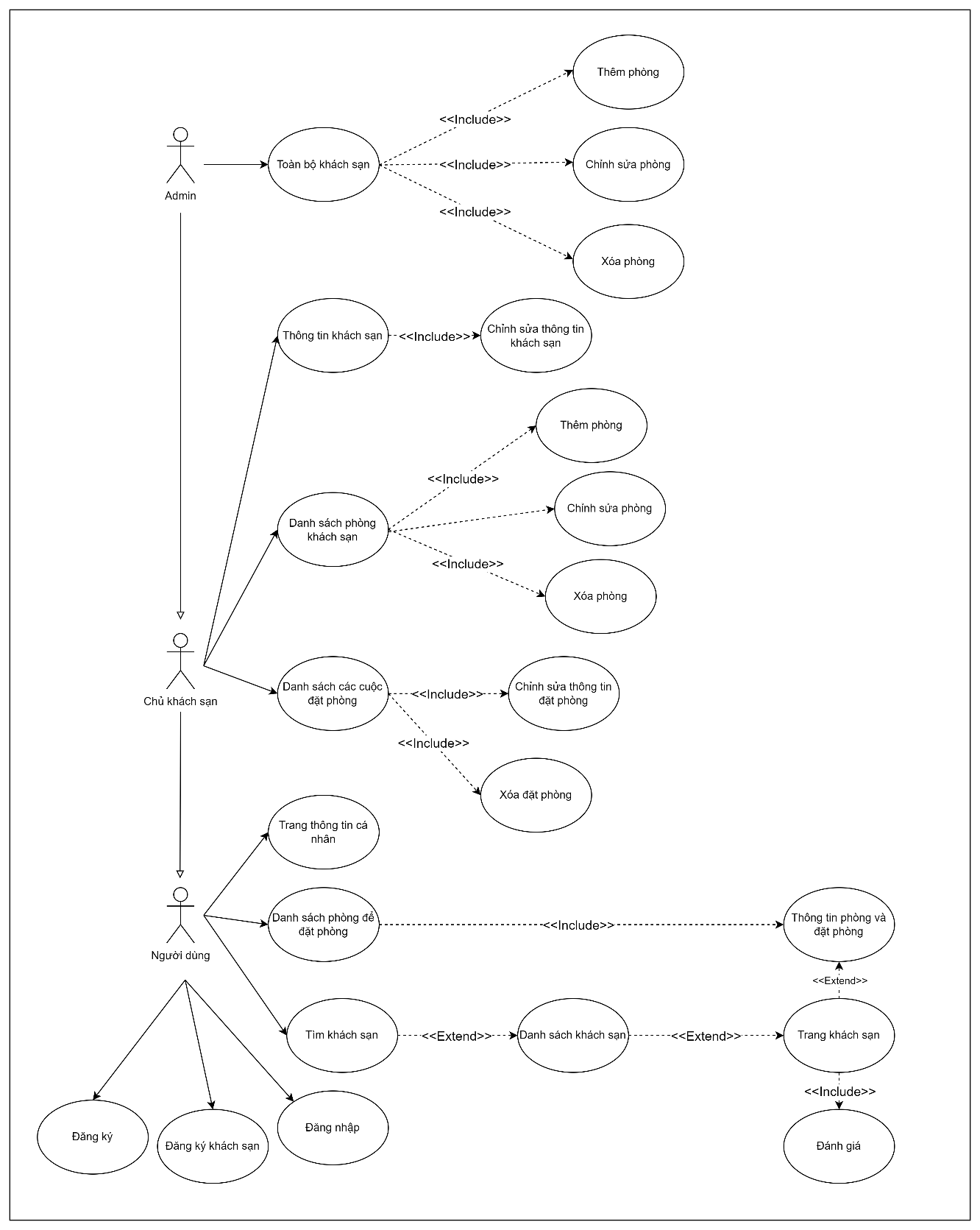
## 1.3. Glossary

|  |  |  |
| --- | --- | --- |
| STT | Thuật ngữ | Giải thích |
| 1 | CSDL | Cơ sở dữ liệu |
| 2 | RDBMS | Relational Database Management System (Hệ quản trị cơ sở dữ liệu quan hệ) |
| 3 | View calendar | Xem lịch làm việc |
| 4 | Timekeeping | Chấm công |
| 5 | View Timekeeping history | Xem lịch sử chấm công |
| 6 | View restaurant list | Xem danh sách nhà hàng |
| 7 | View menu | Xem thực đơn |
| 8 | Edit Menu | Chỉnh sửa thực đơn |
| 9 | Order | Đặt món |
| 10 | Check the stock | Kiểm tra kho hàng |
| 11 | View table list | Xem danh sách bàn |
| 12 | Reservation | Đặt bàn |
| 13 | View bill history | Xem lịch sử hóa đơn |
| 14 | Login | Đăng nhập |
| 15 | Exchange points | Đổi điểm |
| 16 | pay the bill | Thanh toán |

Bảng 1: Glossary

# CHƯƠNG 2: USE CASE MODEL

## 2.1. Use case diagram



Hình 1: Use case diagram

## 2.2. Use case specification / scemarios

### 2.2.1. Use case Đăng ký

|  |  |
| --- | --- |
| Use case | Đăng ký |
| Brief description | Cho phép người dùng đăng ký tài khoản |
| Actors | Người dùng |
| Pre-conditions | Chức năng này chỉ có thể thực hiện sau khi người dùng điền đầy đủ thông tin hợp lệ vô hệ thống |
| Post-conditions | Khi một người dùng muốn đăng ký tài khoản cá nhân  >>Sau khi use case được thực hiện thành công, hệ thống tự động lấy thông tin cá nhân của được nhập và lưu vào hệ thống |
| BasicfFlow | 1. Người dùng nhập thông tin hợp lệ vào trong bảng đăng ký  2. Người dùng nhấn nút đăng ký |
| SubFlows | Không có |
| Alternative flows | Hệ thống kiểm tra không hợp lệ(do không kết nối được với hệ thống)  1. Hệ thống hiển thị thông báo không thể vào thông tin cá nhân với lý do cụ thể |
| Scenarios | S1. BasicfFlow, hệ thống thông báo không hợp lệ |

Bảng 2: Use case đăng ký

### 2.2.2. Use case đăng ký khách sạn

|  |  |
| --- | --- |
| Use case | Đăng ký khách sạn |
| Brief description | Cho phép người dùng đăng khách sạn |
| Actors | Người dùng |
| Pre-conditions | Chức năng này chỉ có thể thực hiện sau khi người dùng điền đầy đủ thông tin hợp lệ vô hệ thống |
| Post-conditions | Khi người dùng muốn đăng ký khách sạn của mình lên trang web  >>Sau khi use case được kích hoạt thành công, hệ thống sẽ tự động lấy thông tin về người dùng và khách sạn để lưu vào hệ thống |
| BasicfFlow | 1. Người dùng nhấn vào nút đăng phòng khách sạn  2. Người dùng nhập toàn bộ thông tin cần thiết và hợp lệ vào hệ thống  3. Người dùng nhấn nút đăng ký khách sạn |
| SubFlows | Không có |
| Alternative flows | Hệ thống kiểm tra không hợp lệ(do không kết nối được với hệ thống)  1. Hệ thống hiển thị thông báo không thể đăng ký khách sạn với lý do cụ thể |
| Scenarios | S1. BasicfFlow, Hệ thống hiển thị thông báo không thể đăng ký khách sạn với lý do cụ thể |

Bảng 3 : Use case đăng ký khách sạn

### 2.2.3. Use case đăng nhập

|  |  |
| --- | --- |
| Use case | Đăng nhập |
| Brief description | Cho người dùng ở bất kì vai trò nào đăng nhập vô tài khoản của mình |
| Actors | Tất cả người dùng |
| Pre-conditions | Chức năng chỉ có thể thực hiện sau khi nhân viên đã đăng nhập thành công vào hệ thống |
| Post-conditions | Khi người dùng cần đăng nhập vào tài khoản  >>Sau khi use case được kích hoạt thành công, hệ thống sẽ tự động lấy thông tin về lịch làm việc của nhân viên đó và hiển thị trên giao diện |
| BasicfFlow | 1. Người dùng bấm vào button đăng nhập trên trang chủ  2. Người dùng nhập toàn bộ thông tin cần thiết  3. Người dùng nhấn nút đăng nhập |
| SubFlows | Không có |
| Alternative flows | Hệ thống kiểm tra không hợp lệ(do không kết nối được với hệ thống)  1. Hệ thống hiển thị thông báo không thể đăng nhập với lý do cụ thể. |
| Scenarios | S1. BasicfFlow, hệ thống đăng nhập không hợp lệ |

Bảng 4: Use case đăng nhập

### 2.2.4. Use case tìm khách sạn

|  |  |
| --- | --- |
| Use case | Tìm khách sạn |
| Brief description | Cho phép mọi người có thể tìm khách sạn dựa vào dữ kiện cụ thể |
| Actors | Toàn bộ người dùng |
| Pre-conditions | Chức năng chỉ có thể thực hiện sau khi người dùng vô được trang web |
| Post-conditions | Khi có một người dùng một tìm khách sạn ở một địa điểm cụ thể  >>Sau khi use case được thực hiện thành công, hệ thống sẽ xuất ra danh sách khách sạn người dùng nhập vào |
| BasicfFlow | 1. Người dùng nhập thông tin ở thanh Search ngay header  2. Người dùng nhấn nút tìm |
| SubFlows | Không có |
| Alternative flows | Hệ thống kiểm tra không hợp lệ(do không kết nối được với hệ thống, đang trong khoảng thời gian nằm ngoài giờ chấm công)  1. Hệ thống hiển thị thông báo không thể chấm công với lý do cụ thể |
| Scenarios | S1. BasicfFlow, hệ thống thông báo không thể tìm được phòng |

Bảng 5: Use case tìm khách sạn

### 2.2.5. Use case danh sách các khách sạn

|  |  |
| --- | --- |
| Use case | Danh sách các khách sạn |
| Brief description | Cho phép những người dùng có thể xem danh sách các khách sạn |
| Actors | Toàn bộ người dùng |
| Pre-conditions | Chức năng chỉ có thể thực hiện sau khi người dùng nhập thông tin vào trong thanh tìm kiếm khách sạn và nhấn tìm kiếm |
| Post-conditions | Khi người dùng muốn tìm kiếm những khách sạn dựa vào những thông tin cụ thể  >>Sau khi use case được thực hiện thành công, hệ thống tự động lấy toàn bộ khách sạn dựa trên thông tin được nhập vào thành tìm kiếm |
| BasicfFlow | 1. Người dùng điền thông tin khách sạn muốn tìm vào thanh tìm kiếm  2. Người dùng nhấn nút tìm |
| SubFlows | Không có |
| Alternative flows | Hệ thống kiểm tra không hợp lệ(do không kết nối được với hệ thống, đang trong khoảng thời gian nằm ngoài giờ chấm công)  1. Hệ thống hiển thị thông báo không thể hiển thị danh sách phòng với lý do cụ thể |
| Scenarios | S1. BasicfFlow, hệ thống thông báo xem lịch sử chấm công không hợp lệ |

Bảng 6: Use case danh sách các khách sạn

### 2.2.6. Use case trang chủ khách sạn

|  |  |
| --- | --- |
| Use case | Trang chủ khách sạn |
| Brief description | Cho phép người dùng xem trang chủ khách sạn |
| Actors | Toàn bộ người dùng |
| Pre-conditions | Chức năng chỉ có thể thực hiện sau khi người dùng nhấn vào một khách sạn cụ thể |
| Post-conditions | Khi một người dùng muốn xem một khách sạn cụ thể  >>Sau khi use case được thực hiện thành công, hệ thống tự động lấy thông tin khách sạn từ hệ thống và hiển thị lên mành ình |
| BasicfFlow | 1. Người dùng nhấn vào khách sạn hiển thị trên màn hình |
| SubFlows | Không có |
| Alternative flows | Hệ thống kiểm tra không hợp lệ(do không kết nối được với hệ thống, đang trong khoảng thời gian nằm ngoài giờ chấm công)  1. Hệ thống hiển thị thông báo không thể lấy thông tin khách sạn với lý do cụ thể  2. Hệ thống hiển thị màn hình chính |
| Scenarios | S1. BasicfFlow, hệ thống thông báo xem khách sạn không hợp lệ |

Bảng 7: Use case trang chủ khách sạn

### 2.2.7. Use case đặt phòng

|  |  |
| --- | --- |
| Use case | Đặt phòng |
| Brief description | Cho phép người dùng đặt phòng với những thông tin cụ thể |
| Actors | Tất cả người dùng |
| Pre-conditions | Chức năng chỉ có thể thực hiện sau khi người dùng điền đầy đủ thông tin phù hợp và nhấn nút đặt phòng |
| Post-conditions | Khi một người dùng muốn đặt phòng  >>Sau khi use case được thực hiện thành công, hệ thống sẽ lưu thông tin đặt phòng vào trong hệ thống và thông báo đặt phòng thành công |
| BasicfFlow | 1. Người dùng bấm vào một khách sạn cụ thể  2. Người dùng bấm vào nút đặt ở một phòng cụ thể  3. Người dùng điền đầy đủ thông tin thích hợp  4. Người dùng nhấn nút đặt phòng |
| SubFlows | Không có |
| Alternative flows | Hệ thống kiểm tra không hợp lệ(do không kết nối được với hệ thống, phòng đã có người đặt trong thời gian này)  1. Hệ thống hiển thị thông báo không thể đặt phòng với lý do cụ thể |
| Scenarios | S1. BasicfFlow, hệ thống thông báo xem thực đơn không hợp lệ |

Bảng 8: Use case đặt phòng

### 2.2.8. Use case danh sách các phòng

|  |  |
| --- | --- |
| Use case | Danh sách các phòng |
| Brief description | Cho phép người dùng xem danh sách các phòng thuộc về một khách sạn cụ thể |
| Actors | Tất cả người dùng |
| Pre-conditions | Chức năng chỉ có thể thực hiện sau người dùng bấm vào một khách sạn cụ thể |
| Post-conditions | Khi một người dùng muốn xem thông tin khách sạn và những phòng có bên trong  >>Sau khi use case được thực hiện thành công, hệ thống tự động lưu những thông tin về phòng thuộc khách sạn đó |
| BasicfFlow | 1. Người dùng bấm vào một khách sạn cụ thể |
| SubFlows | Không có |
| Alternative flows | Hệ thống kiểm tra không hợp lệ(do không kết nối được với hệ thống)  1. Hệ thống hiển thị thông báo không thể hiển thị phòng với lý do cụ thể |
| Scenarios | S1. BasicfFlow, hệ thống thông báo xem có phòng không |

Bảng 9: Use case danh sách các phòng

### 2.2.9. Use case thông tin các nhân

|  |  |
| --- | --- |
| Use case | Thông tin cá nhân |
| Brief description | Cho phép người dùng xem thông tin cá nhân |
| Actors | Tất cả người dùng |
| Pre-conditions | Chức năng chỉ có thể thực hiện sau khi người dùng đã đăng nhập thành công vào hệ thống |
| Post-conditions | Khi một người dùng muốn xem thông tin của tài khoản của mình  >>Sau khi use case được thực hiện thành công, hệ thống tự động lấy thông tin từ hệ thống và hiển thị lên màn hình |
| BasicfFlow | 1. Người dùng nhấn nút đăng nhập  2. Người dùng nhập thông tin cần thiết để đăng nhập và nhấn nút đăng nhập  3. Người dùng nhấn vào biểu tượng profile và bấm tài khoản |
| SubFlows | Không có |
| Alternative flows | Hệ thống kiểm tra không hợp lệ(do không kết nối được với hệ thống)  1. Hệ thống hiển thị thông báo không thể hiển thị phòng với lý do cụ thể |
| Scenarios | S1. BasicfFlow, người dùng bấm vào tài khoản |

Bảng 10: Use case thông tin cá nhân

### 2.2.10. Use case xem đánh giá

|  |  |
| --- | --- |
| Use case | Xem đánh giá |
| Brief description | Cho phép người dùng xem đánh giá của khách sạn cụ thể |
| Actors | Tất cả người dùng |
| Pre-conditions | Chức năng chỉ có thể thực hiện khi người dùng bấm vào khách sạn cụ thể |
| Post-conditions | Khi một người dùng cần xem đánh giá của một khách sạn  >> Sau khi use case được thực hiện thành công, hệ thống tự động lấy thông tin về những đánh giá thuộc khách sạn đó và hiển thị lên màn hình |
| BasicfFlow | 1. Người dùng nhấn vào một khách sạn cụ thể  2. Người dùng nhấn vào nút xem đánh giá |
| SubFlows | Không có |
| Alternative flows | Hệ thống kiểm tra không hợp lệ(do không kết nối được với hệ thống)  1. Hệ thống hiển thị thông báo không thể xem đánh giá với lý do cụ thể |
| Scenarios | S1. BasicfFlow, hệ thống thông báo không thể xem đánh giá |

Bảng 11: Use case xem đánh giá

### 2.2.11. Use case đánh giá

|  |  |
| --- | --- |
| Use case | Đánh giá |
| Brief description | Cho phép người dùng đánh giá phòng khách sạn sau khi đã đặt xong |
| Actors | Tất cả người dùng |
| Pre-conditions | Chức năng chỉ có thể thực hiện sau khi người dùng đã đặt phòng |
| Post-conditions | Khi một người dùng muốn đặt phòng  >> Sau khi use case được thực hiện thành công, hệ thống tự động lấy thông tin từ đánh giá đó và lưu vào hệ thống |
| BasicfFlow | 1. Người dùng bấm vào khách sạn  2. Người dùng bấm đặt phòng và đặt một phòng  3. Người dùng sau khi đặt xong bấm vào xem đánh giá  4. Người dùng nhập đánh giá và nhấn nút gửi |
| SubFlows | Không có |
| Alternative flows | Hệ thống kiểm tra không hợp lệ(do không kết nối được với hệ thống)  1. Hệ thống hiển thị thông báo không thể bình luận với lý do cụ thể |
| Scenarios | S1. BasicfFlow, hệ thống thông báo bình luận không hợp lệ |

Bảng 12: Use case đánh giá

### 2.2.12. Use case thêm phòng

|  |  |
| --- | --- |
| Use case | Thêm phòng |
| Brief description | Cho phép chủ khách sạn và admin thêm phòng |
| Actors | Chủ khách sạn / Admin |
| Pre-conditions | Chức năng chỉ có thể thực hiện sau khi chủ khách sạn và quản trị viên đã đăng nhập thành công vào hệ thống |
| Post-conditions | Khi chủ khách sạn và quản trị viên muốn thêm phòng vào khách sạn  >> Sau khi use case được thực hiện thành công, hệ thống tự động lấy thông tin bàn từ phòng mới vừa nhập vào và thêm vào hệ thống |
| BasicfFlow | 1. Chủ khách sạn / Admin bấm vào trang quản lý phòng  2. Chủ khách sạn / Admin nhấn vào nút thêm phòng  3. Chủ khách sạn / Admin nhập thông tin phòng và bấm vào nút thêm phòng |
| SubFlows | Không có |
| Alternative flows | Hệ thống kiểm tra không hợp lệ(do không kết nối được với hệ thống)  1. Hệ thống hiển thị thông báo không thể thêm phòng với lý do cụ thể |
| Scenarios | S1. BasicfFlow, hệ thống thông báo thêm phòng không hợp lệ |

Bảng 13: Use case thêm phòng

### 2.2.13. Use case chỉnh sửa phòng

|  |  |
| --- | --- |
| Use case | Chỉnh sửa phòng |
| Brief description | Cho phép chủ khách sạn / Admin thay đổi thông tin của một phòng bất kì |
| Actors | Chủ khách sạn / Admin |
| Pre-conditions | Chức năng chỉ có thể thực hiện sau khi chủ khách sạn / Admin đã đăng nhập vào hệ thống |
| Post-conditions | Khi chủ khách sạn / Admin muốn thay đổi thông tin của một phòng có sẵn  >>Sau khi use case kích hoạt thành công thì hệ thống sẽ tự động lấy thông tin về toàn bộ thông tin đã thay đổi và lưu vào hệ thống |
| BasicfFlow | 1. Chủ khách sạn / Admin bấm vào trang quản lý phòng  2. Chủ khách sạn / Admin nhấn vào nút sửa phòng ở một phòng cụ thể.  3. Chủ khách sạn / Admin nhập thông tin cần thay đổi và nhấn nút xác nhận |
| SubFlows | Không có |
| Alternative flows | Hệ thống kiểm tra xem chỉnh sửa phòng không hợp lệ(không kết nối được với hệ thống)  1. Hệ thống hiển thị thông báo không thể chỉnh sửa phòng với lý do cụ thể |
| Scenarios | S1. BasicfFlow, hệ thống kiểm tra xem chỉnh sửa phòng không hợp lệ |

Bảng 14: Use case chỉnh sửa phòng

### 2.2.14. Use case xóa phòng

|  |  |
| --- | --- |
| Use case | Xóa phòng |
| Brief description | Cho phép chủ khách sạn / Admin xóa phòng |
| Actors | Chủ khách sạn / Admin |
| Pre-conditions | Chức năng chỉ có thể thực hiện khi có kết nối mạng và có phòng để xóa |
| Post-conditions | Khi chủ khách sạn / Admin đăng nhập vào phần mềm  >>Sau khi use case được kích hoạt thành công, hệ thống sẽ xóa phòng được chọn |
| BasicfFlow | 1. Chủ khách sạn / Admin bấm vào trang quản lý phòng  2. Chủ khách sạn / Admin nhấn vào nút xóa phòng ở một phòng cụ thể  3. Chủ khách sạn / Admin nhấn nút xác nhận |
| SubFlows | Không có |
| Alternative flows | Hệ thống kiểm tra không hợp lệ(do không kết nối được với hệ thống)  1. Hệ thống hiển thị thông báo không thể xóa phòng với lý do cụ thể |
| Scenarios | S1. BasicfFlow, hệ thống kiểm tra xem xóa phòng có hợp lệ không |

Bảng 15: Use case xóa phòng

### 2.2.15. Use case xem danh sách đặt phòng

|  |  |
| --- | --- |
| Use case | Xem danh sách đặt phòng |
| Brief description | Cho phép chủ khách sạn / Admin xem danh sách đặt phòng |
| Actors | Chủ khách sạn / Admin |
| Pre-conditions | Chức năng chỉ có thể thực hiện khi có kết nối mạng và chủ khách sạn hoặc Admin đăng nhập |
| Post-conditions | Chủ khách sạn / Admin đăng nhập vào phần mềm  >>Sau khi use case được kích hoạt thành công, hệ thống sẽ cho lấy những phòng được đặt tại khách sạn và xuất ra màn hình |
| BasicfFlow | 1. Chủ khách sạn / Admin nhấn vào nút quản lý đặt phòng |
| SubFlows | Không có |
| Alternative flows | Hệ thống kiểm tra không hợp lệ(do không kết nối được với hệ thống)  1. Hệ thống hiển thị thông báo không thể xem danh sashc đặt phòng với lý do cụ thể |
| Scenarios | S1. BasicfFlow, hệ thống kiểm tra xem danh sách đặt phòng không hợp lệ |

Bảng 16: Use case xem danh sách đặt phòng

### 2.2.16. Use case chỉnh sửa thông tin đặt phòng

|  |  |
| --- | --- |
| Use case | Chỉnh sửa thông tin đặt phòng |
| Brief description | Cho phép chủ khách sạn / Admin thay đổi thông tin của phòng được đặt |
| Actors | Chủ khách sạn / nhân viên |
| Pre-conditions | Chức năng chỉ có thể thực hiện sau khi chủ khách sạn / Admin đã đăng nhập thành công vào hệ thống |
| Post-conditions | Chủ khách sạn / Admin đăng nhập vào web  >>Sau khi use case chạy thành công hệ thống sẽ tự động cập toàn bộ thông tin đặt phòng được thay đổi vào hệ thống |
| BasicfFlow | 1. Chủ khách sạn / Admin bấm vào trang quản lý đặt phòng  2. Chủ khách sạn / Admin nhấn vào nút chỉnh sửa phòng ở một phòng cụ thể thể  3. Chủ khách sạn / Admin nhập đầy đủ thông tin và nhấn nút xác nhận |
| SubFlows | Không có |
| Alternative flows | Hệ thống kiểm tra thay đổi phòng không hợp lệ(không kết nối được với hệ thống)  1. Hệ thống hiển thị thông báo không thể thay đổi phòng với lý do cụ thể |
| Scenarios | S1. BasicfFlow, hệ thống kiểm tra việc cập nhật phòng không hợp lệ |

Bảng 17: Use case chỉnh sửa thông tin phòng

### 2.2.17. Use case xóa đặt phòng

|  |  |
| --- | --- |
| Use case | Xóa đặt phòng |
| Brief description | Cho phép chủ khách sạn / Admin đăng ký tài khoản |
| Actors | Chủ khách sạn / Admin |
| Pre-conditions | Chức năng chỉ có thể thực hiện khi chủ khách sạn / Admin đăng nhập |
| Post-conditions | Khi chủ khách sạn / Admin cần xóa một phòng đã đặt  >>Sau khi use case được kích hoạt thành công, hệ thống sẽ xóa phòng đã đặt đó |
| BasicfFlow | 1. Chủ khách sạn / Admin bấm vào trang quản lý đặt phòng  2. Chủ khách sạn / Admin nhấn vào nút hủy ở một phòng cụ thể thể  3. Chủ khách sạn / Admin nhấn nút xác nhận |
| SubFlows | Không có |
| Alternative flows | Hệ thống kiểm tra không hợp lệ(do không kết nối được với hệ thống)  1. Hệ thống hiển thị thông báo không thể xóa với lý do cụ thể |
| Scenarios | S1. BasicfFlow, hệ thống kiểm tra xem việc xóa không hợp lệ |

Bảng 18: Use case xóa đặt phòng

### 2.2.18. Use case chỉnh sửa thông tin khách sạn

|  |  |
| --- | --- |
| Use case | Chỉnh sửa thông tin khách sạn |
| Brief description | Cho chủ khách sạn / Admin chỉnh sửa lại thông tin của khách sạn |
| Actors | Chủ khách sạn/ Admin |
| Pre-conditions | Chức năng chỉ có thể thực hiện khi có kết nối mạng |
| Post-conditions | Khi chủ khách sạn / Admin muốn chỉnh sửa lại thông tin khách sạn  >>Sau khi use case được kích hoạt thành công, hệ thống sẽ lấy toàn bộ thông tin vừa chỉnh sửa và lưu lại |
| BasicfFlow | 1. Chủ khách sạn / Admin nhấn vào trang thông tin khách sạn.  2. Chủ khách sạn / Admin nhấn |
| SubFlows | Không có |
| Alternative flows | Hệ thống kiểm tra không hợp lệ(do không kết nối được với hệ thống)  1. Hệ thống hiển thị thông báo không thể chỉnh sửa phòng với lý do cụ thể |
| Scenarios | S1. BasicfFlow, hệ thống kiểm tra xem chỉnh sửa phòng không hợp lệ |

Bảng 19: Use case chỉnh sửa phòng

# CHƯƠNG 3: SYSTEM ARCHITECT

## 3.1.Phatform

Web: được chọn làm phatform phát triển vì khả năng tiếp cận rộng rãi, Web ứng dụng có thể truy cập từ bất kỳ thiết bị nào có trình duyệt và kết nối internet, bao gồm máy tính, điện thoại, máy tính bảng. Người dùng không cần phải tải và cài đặt ứng dụng, giúp tiết kiệm thời gian và không gian lưu trữ. Cập nhật và bảo trì ứng dụng web dễ dàng hơn vì chỉ cần cập nhật trên máy chủ và tất cả người dùng sẽ có phiên bản mới nhất khi truy cập. Phát triển một ứng dụng web thường rẻ hơn so với việc phát triển ứng dụng gốc (native app) cho nhiều nền tảng khác nhau như iOS và Android. Ứng dụng web có thể hoạt động trên nhiều hệ điều hành và thiết bị khác nhau mà không cần phải thay đổi mã nguồn nhiều. Các công nghệ web hiện nay như React, Angular, Vue.js giúp tăng tốc độ phát triển và triển khai ứng dụng. Ứng dụng web có thể dễ dàng tích hợp với nhiều dịch vụ và công cụ khác qua API và các giao thức web. Ứng dụng web có khả năng mở rộng dễ dàng hơn để phục vụ một lượng lớn người dùng.

## 3.2.Lựa chọn ngôn ngữ

Java: vì lJava có rất nhiều tính năng bảo mật có sẵn cần thiết cho lập trình web như: Xác thực nâng cao, mã hóa và kiểm soát mức độ truy cập. Do tính chất chặt chẽ của ngôn ngữ Java. Bạn có thể kiểm soát từng dòng code mình viết ra. Chương trình của bạn về cơ bản đã tự có tính bảo mật cao ngay cả khi chưa áp dụng biện pháp bảo mật kèm theo nào.

## 3.3.Lựa chọn framework và thư viện

### 3.3.1. Giới thiệu về Reactjs

ReactJS được phát triển bởi Facebook và được giới thiệu lần đầu tiên vào năm 2011 nhằm cải thiện tốc độ và hiệu suất của ứng dụng web.

Với những tiện ích cho việc phát triển ứng dụng web, bao gồm:

- Hiệu suất cao: ReactJS sử dụng Virtual DOM để tối ưu hóa hiệu suất của ứng dụng. Virtual DOM cho phép ReactJS cập nhật các thay đổi trên trang web một cách nhanh chóng và hiệu quả hơn so với cách truyền thống, giúp tăng tốc độ và hiệu suất của ứng dụng.

- Tái sử dụng: ReactJS cho phép tái sử dụng các thành phần UI, giúp giảm thiểu thời gian và chi phí phát triển. Các thành phần UI có thể được sử dụng lại trong nhiều phần khác nhau của ứng dụng, giúp tăng tính linh hoạt và khả năng mở rộng của ứng dụng.

- Dễ dàng quản lý trạng thái: ReactJS giúp quản lý trạng thái của ứng dụng một cách dễ dàng. Sử dụng State và Props, ReactJS cho phép các nhà phát triển quản lý trạng thái của các thành phần UI một cách chính xác và dễ dàng.

- Hỗ trợ tốt cho SEO: ReactJS cho phép các nhà phát triển xây dựng ứng dụng web với khả năng tương thích tốt với SEO. Với sự hỗ trợ của các thư viện như React Helmet, ReactJS cho phép các nhà phát triển tùy chỉnh và quản lý các phần tử meta và title cho từng trang web.

- Hỗ trợ đa nền tảng: ReactJS không chỉ được sử dụng để phát triển các ứng dụng web, mà còn được sử dụng để phát triển các ứng dụng di động với React Native. Sử dụng React Native, các nhà phát triển có thể xây dựng ứng dụng di động cho cả iOS và Android sử dụng cùng một mã nguồn.

### 3.3.2. Giới thiệu về Spring boot

Spring Boot là một framework Java được sử dụng để xây dựng các ứng dụng và dịch vụ web dễ dàng và nhanh chóng. Nền tảng cung cấp các cấu hình mặc định cho một số thư viện và bộ công cụ hỗ trợ xây dựng, triển khai, quản lý ứng dụng Spring-based.

Cách Spring Boot hoạt động nhằm tối ưu hóa quy trình phát triển ứng dụng Java. Điều này sẽ giúp nhà phát triển tập trung vào việc xây dựng tính năng chính của ứng dụng mà không cần phải lo lắng về cấu hình phức tạp.

Một số ưu điểm dưới đây đã giúp Spring Boot trở thành sự lựa chọn phổ biến cho việc xây dựng các ứng dụng và dịch vụ web trong hệ sinh thái Java:

- Tối ưu hóa quá trình phát triển: Spring Boot cung cấp cấu hình mặc định thông minh và tự động, giúp giảm thiểu việc cấu hình thủ công và tối ưu quá trình phát triển ứng dụng Java.

- Tích hợp tốt: Spring Boot tích hợp tốt với nhiều công nghệ và thư viện khác trong hệ sinh thái Spring Framework. Nền tảng cho phép hệ thống dễ dàng tích hợp các module và dịch vụ khác nhau mà không cần phải lo lắng về cấu hình phức tạp.

- Embedded server: Spring Boot đi kèm với các máy chủ nhúng như Tomcat, Jetty, hoặc Undertow. Đây là công cụ không thể thiếu trong việc triển khai ứng dụng một cách đơn giản mà không cần cấu hình thêm bất kỳ máy chủ nào khác.

- Tự động cấu hình: Spring Boot sử dụng cơ chế cấu hình tự động thông minh, cho phép ứng dụng tự cấu hình dựa trên các thư viện và module được sử dụng.

- Quản lý phụ thuộc: Spring Boot cung cấp các công cụ quản lý phụ thuộc mạnh mẽ như Maven hoặc Gradle, giúp quản lý các phụ thuộc của ứng dụng một cách hiệu quả.

- Monitoring và quản lý: Spring Boot cung cấp các công cụ hỗ trợ giám sát và quản lý ứng dụng dễ dàng, bao gồm Spring Boot Actuator cho việc giám sát và quản lý ứng dụng.

### 3.3.3. Giới thiệu Spring Security

Spring Security được phát triển bởi SpringSource (hiện thuộc Pivotal) và được xem là một trong những framework bảo mật phổ biến nhất cho ứng dụng Java. Spring Security được thiết kế theo kiến trúc plugin, cho phép tùy biến linh hoạt và dễ dàng theo nhu cầu của ứng dụng và được tích hợp sẵn với các thành phần khác của Spring Framework, như Spring Boot, Spring MVC, Spring Data, Spring Cloud, và Spring WebFlux.

Spring Security hoạt động theo mô hình client-server. Khi một client gửi một request đến server, server sẽ xác thực người dùng và phân quyền để đảm bảo rằng người dùng chỉ có thể truy cập vào những tài nguyên mà họ được phép truy cập.

Cơ chế hoạt động của Spring Security dựa trên cơ chế lọc (filter) và sự kiện (event) để can thiệp vào quá trình xử lý yêu cầu (request) và phản hồi (response) của ứng dụng web, tức là khi một yêu cầu được gửi đến ứng dụng web, nó sẽ được chuyển qua một chuỗi các bộ lọc (filter chain) do Spring Security quản lý. Mỗi bộ lọc có một nhiệm vụ cụ thể, như kiểm tra xác thực, kiểm tra phân quyền, điều hướng đến trang đăng nhập hoặc đăng xuất, xử lý các lỗi bảo mật.

Nếu một yêu cầu không thỏa mãn các điều kiện bảo mật của ứng dụng, Spring Security sẽ sinh ra một sự kiện (event) để thông báo cho ứng dụng biết. Ứng dụng có thể lắng nghe và xử lý các sự kiện này theo ý muốn, ví dụ như ghi log, gửi email hoặc hiển thị thông báo lỗi.

Ngược lại, nếu một yêu cầu được chấp nhận bởi Spring Security, nó sẽ được tiếp tục xử lý bởi ứng dụng web như bình thường. Khi ứng dụng web trả về một phản hồi cho yêu cầu, nó cũng sẽ được chuyển qua lại chuỗi các bộ lọc của Spring Security để áp dụng các thiết lập bảo mật cho phản hồi.

### 3.3.4. Giới thiệu Hibernate

Hibernate là một thư viện ORM (Object Relational Mapping) mã nguồn mở giúp lập trình viên viết ứng dụng Java có thể map các objects (pojo) với hệ quản trị cơ sở dữ liệu quan hệ, và hỗ trợ thực hiện các khái niệm lập trình hướng đối tượng với cớ dữ liệu quan hệ.

Hibernate cung cấp các câu lệnh truy vấn tương tự SQL, HQL của Hibernate hỗ trợ đầy đủ các truy vấn đa hình như, HQL “hiểu” các khái niệm như kế thừa (inheritance), đa hình (polymorphysm), và liên kết (association).

Code sử dụng Hibernate là độc lập với hệ quản trị cơ sở dữ liệu, nghĩa là ta không cần thay đổi câu lệnh HQL khi ta chuyển từ hệ quản trị CSDL MySQL sang Oracle, hay các hệ quản trị CSDL khác… Do đó rất dễ để ta thay đổi CSDL quan hệ, đơn giản bằng cách thay đổi thông tin cấu hình hệ quản trị CSDL trong file cấu hình.

Với những ứng dụng Java làm việc với cơ sở dữ liệu lớn hàng trăm triệu bản ghi, việc có sử dụng Lazy loading trong truy xuất dữ liệu từ database mang lại lợi ích rất lớn. Nó giống như việc ta có thể bẻ từng chiếc đũa của bó đũa to thay vì bẻ cả bó đũa.

## 3.4.Lựa chọn Database

MySQL: là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. Người dùng có thể tải về MySQL miễn phí từ trang chủ. MySQL có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS,..

## 3.5.Lựa chọn Deloyment

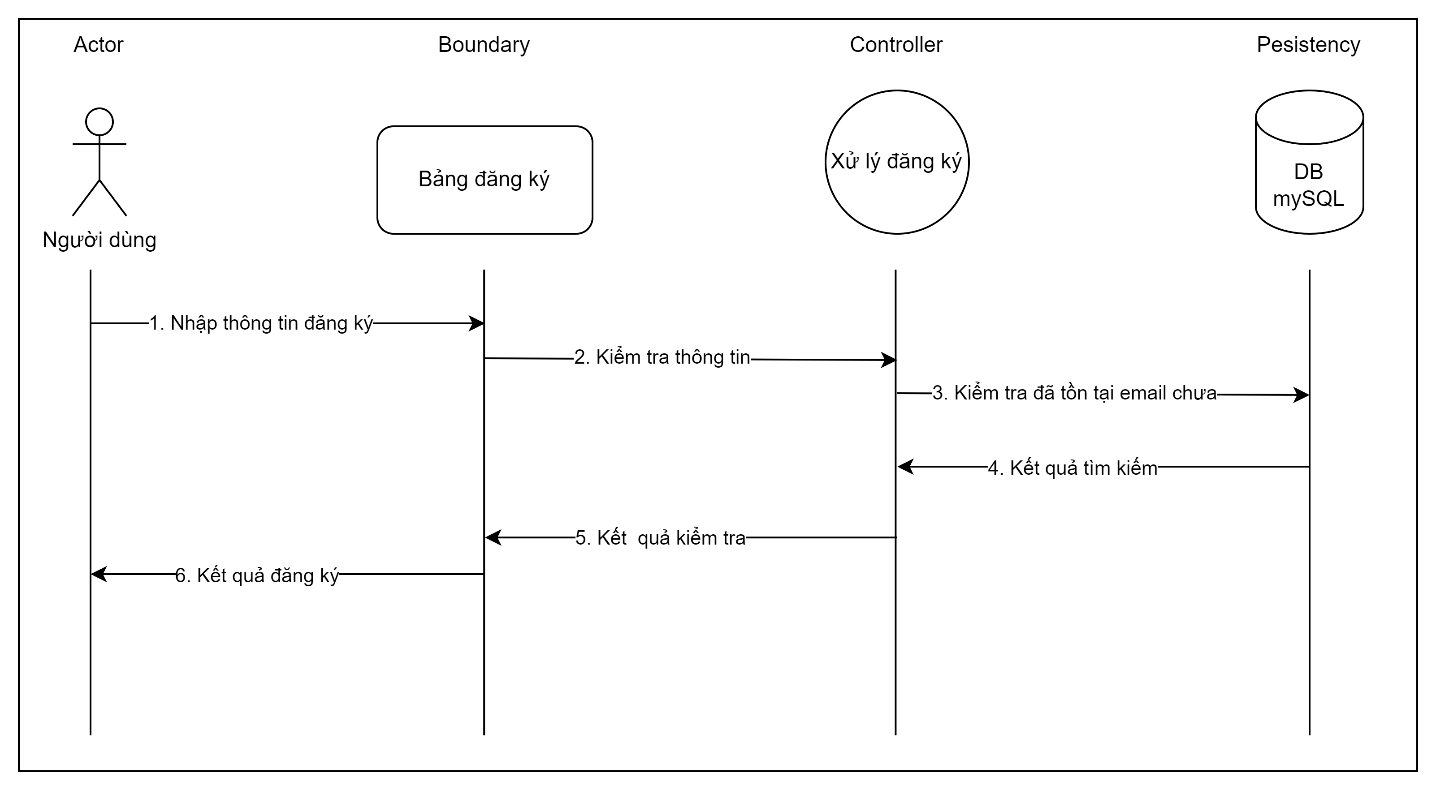
Shared Hosting: là phương pháp lưu trữ website phổ biến, trong đó nhiều trang web chia sẻ tài nguyên của cùng một máy chủ. Khi người dùng truy cập trang web, dữ liệu được gửi từ máy chủ tới họ qua Internet. Máy chủ được tối ưu cho nhiều trang web, nhưng lưu lượng truy cập khác nhau có thể ảnh hưởng đến hiệu suất. Dữ liệu của từng trang web được bảo vệ riêng tư và an toàn trong môi trường chia sẻ này. Shared Hosting cung cấp dịch vụ hỗ trợ và quản lý máy chủ, giúp người dùng quản lý trang web mà không cần kiến thức kỹ thuật sâu.

## 3.6.Lựa chọn Monitoring

Google Analytics: theo dõi hiệu suất, phân tích chiến lực tiếp thị, xem chỉ số tương tác và sử dụng ứng dụng,…

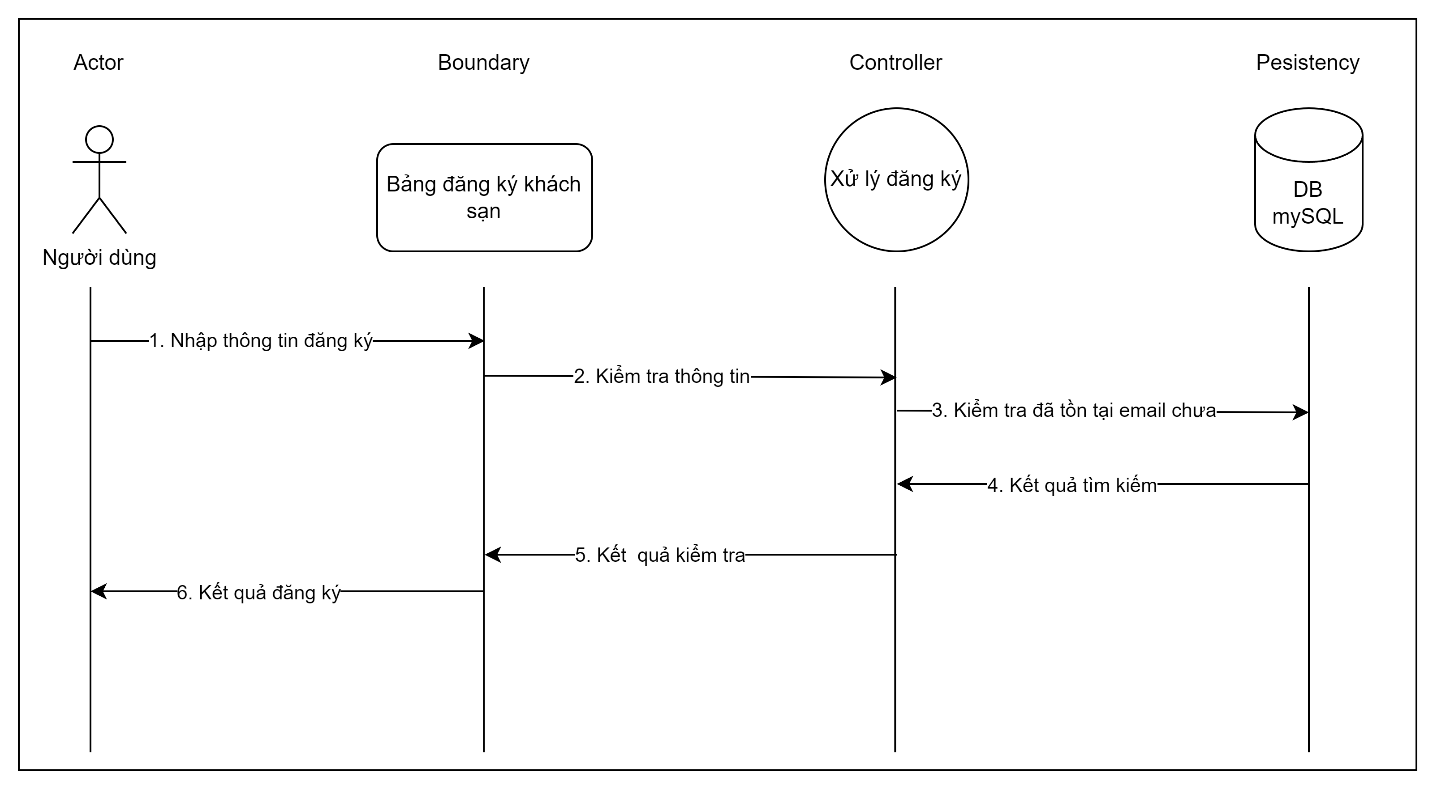
# CHƯƠNG 4: SYSTEM BEHAVIOR

## 4.1. Sequence diagram đăng ký



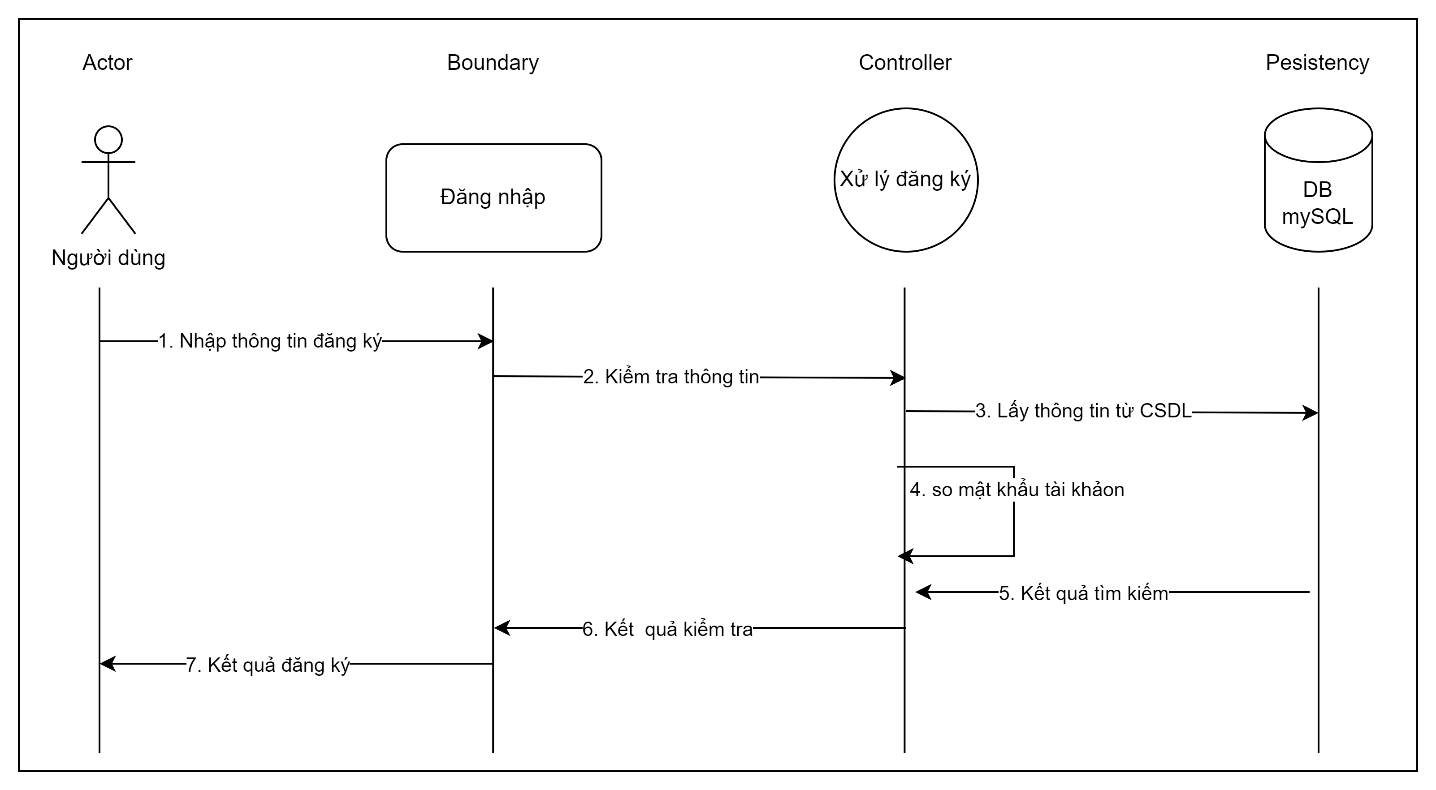
Hình 2: Sequence diagram đăng ký

## 4.2. Sequence diagram Đăng ký khách sạn



Hình 3: Sequence diagram đăng ký khách sạn

## 4.3. Sequence diagram đăng nhập



Hình 4: Sequence diagram đăng nhập

## 4.4. Sequence diagram tìm khách sạn

Hình 5: Sequence diagram tìm khách sạn

## 4.5. Sequence diagram danh sách khách sạn

Hình 6: Sequence diagram danh sách khách sạn

## 4.6. Sequence diagram trang chủ khách sạn

Hình 7: Sequence diagram trang chủ khách sạn

## 4.7. Sequence diagram đặt phòng

Hình 8: Sequence diagram đặt phòng

## 4.8. Sequence diagram danh sách các phòng

Hình 9: Sequence diagram danh sách các phòng

## 4.9. Sequence diagram thông tin cá nhân

Hình 10: Sequence diagram thông tin cá nhân

## 4.10. Sequence diagram xem đánh giá

Hình 11: Sequence diagram xem đánh giá

## 4.11. Sequence diagram đánh giá

Hình 12: Sequence diagram đánh giá

## 4.12. Sequence diagram thêm phòng

Hình 13: Sequence diagram thêm phòng

## 4.13. Sequence diagram chỉnh sửa phòng

Hình 14: Sequence diagram chỉnh sửa phòng

## 4.14. Sequence diagram xóa phòng

Hình 16: Sequence diagram xóa phòng

## 4.15. Sequence diagram xem danh sách những phòng đã đặt

Hình 17: Sequence diagram xem danh sách những phòng đã đặt

## 4.16. Sequence diagram chỉnh sửa thông tin đặt phòng

Hình 18: Sequence diagram chỉnh sửa thông tin đặt phòng

## 4.17. Sequence diagram xóa đặt phòng

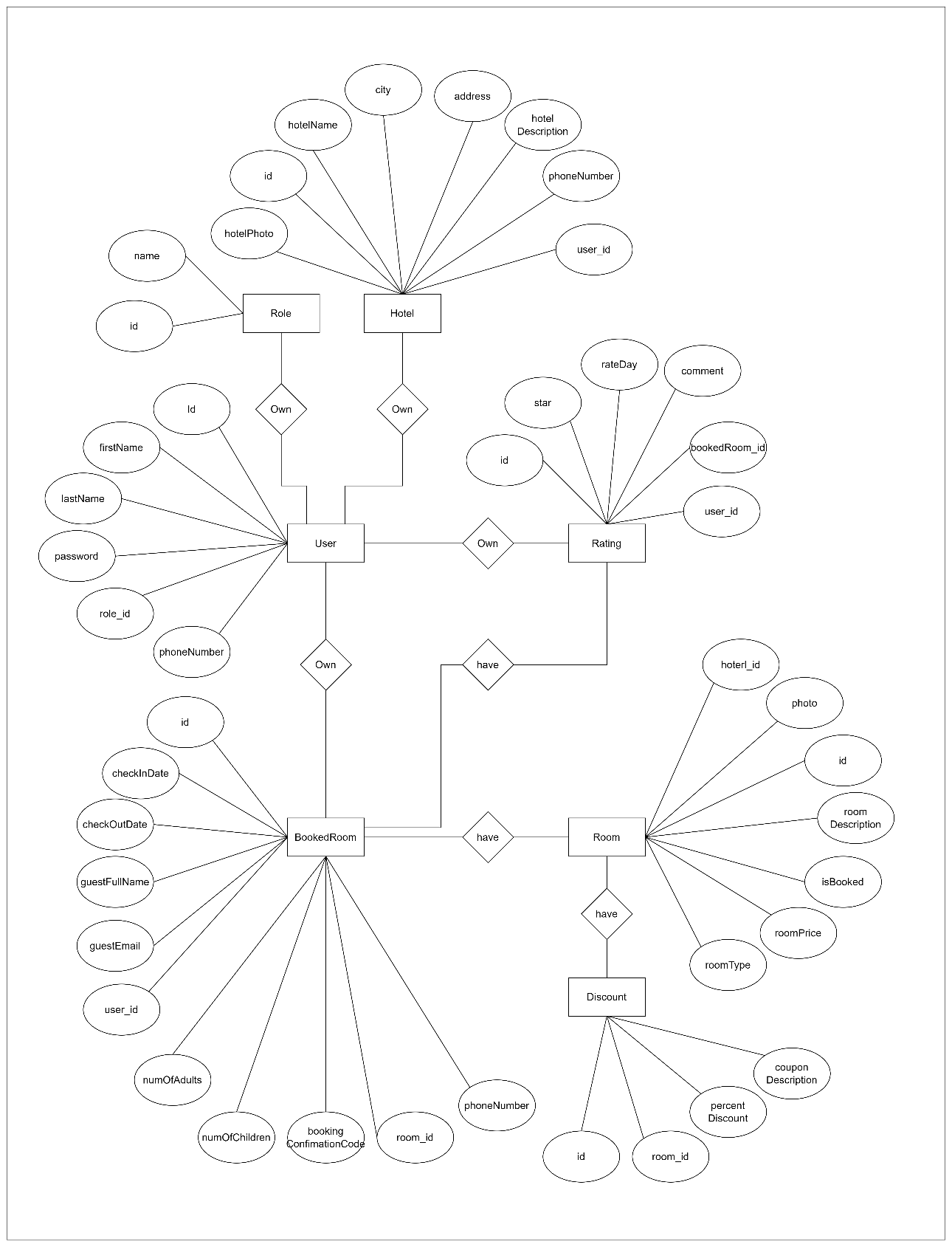
Hình 19: Sequence diagram xóa đặt phòng

## 4.18. Sequence diagram chỉnh sửa thông tin khách sạn

Hình 20: Sequence diagram chỉnh sửa thông tin khách sạn

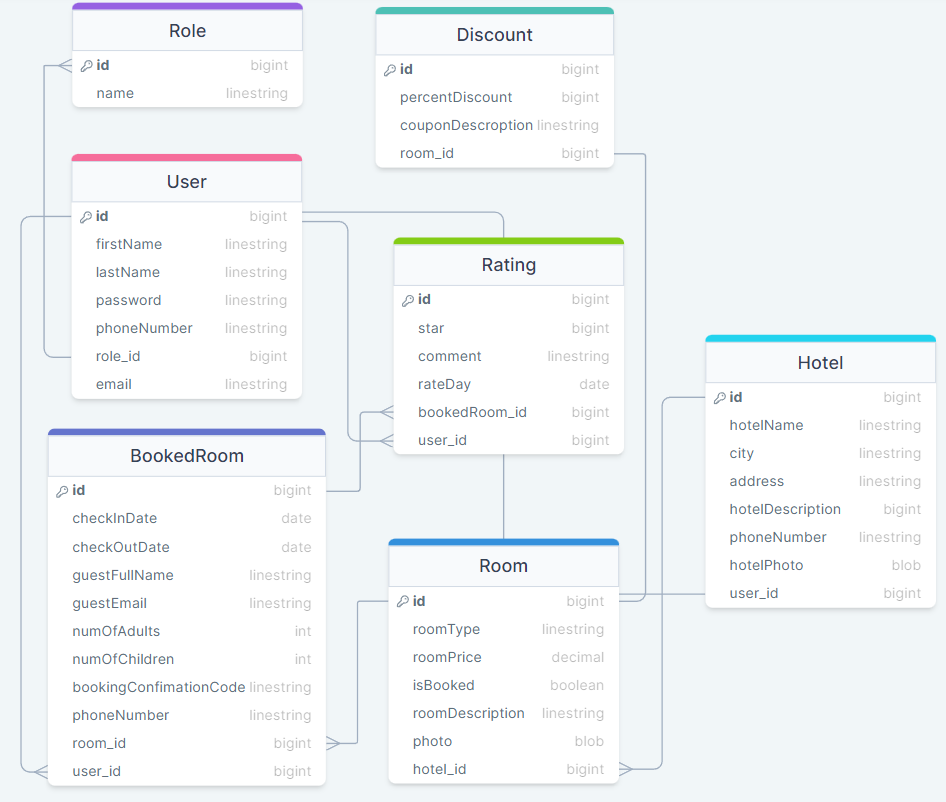
# CHƯƠNG 5: SYSTEM DATA MODEL

## 5.1. Lược đồ thực thể kết gợp(ERD):



Hình 12: Lược đồ thực thể kết gợp(ERD)

## 5.2. Lược đồ CSDL / DataModel



Hình 13: Lược đồ CSDL / DataModel

# TỔNG KẾT

**Kết quả nghiên cứu, tìm hiểu:**

Báo cáo được thực hiện trong một thời gian vừa đủ nhưng vì độ rộng nên vẫn bị giới hạn trong một phạm vi nhất định, song nó vẫn đạt được một số kết quả như sau:

**Các hạn chế và hướng phát triển:**

Hạn chế:

Trong khuôn khổ của bài báo cáo này, việc phân tích hệ thống thông tin bị hạn chế lại ở một quy mô nhất định để tránh quá tải.

Do khối lượng kiến thức lẫn kỹ năng cần thiết để thực hiện báo cáo này là rất nhiều nên báo cáo này sẽ dừng lại ở những thứ cơ bản.

Hiện tại kiến thức và trình độ bản thân còn hạn chế nên báo cáo không thể tránh khỏi những thiếu sót. Em mong rằng sẽ nhận được những ý kiến và đóng góp của thầy để bài báo cáo hoàn thiện hơn, có ích hơn trong thực tế.

Hướng phát triển:

Hướng phát triển nghiên cứu trong tương lai:

Phát triển đồ án ở quy mô rộng hơn kèm theo đó là sự hoàn chỉnh lẫn kiến thức chưa được thể hiện trong báo cáo này.

Phát triển báo cáo thành một phần mềm hoàn chỉnh.

# TÀI LIỆU THAM KHẢO

1. Trang Vũ. (2024). ReactJS là gì? Tất tần tật những điều căn bản về ReactJS. <https://stringee.com/vi/blog/post/reactJS-la-gi>
2. Lê Trí Dũng. (Không rõ thời điểm đăng). Hibernate là gì? Sao phải dùng nó thay JDBC? <https://topdev.vn/blog/hibernate-la-gi-sao-phai-dung-no-thay-jdbc/>
3. Nam Thang. (2023). Spring Security là gì? Cơ chế hoạt động của Spring Security. <https://200lab.io/blog/co-che-hoat-dong-cua-spring-security/>
4. Cao Lê Viết Tiến. (không rõ thời điểm đăng). Tìm hiểu mô hình MVC là gì? Ví dụ về cách sử dụng mô hình MVC. <https://vietnix.vn/tim-hieu-mo-hinh-mvc-la-gi/>
5. Phạm Quí Phương (2021). Tìm hiểu về My SQL. <https://viblo.asia/p/tim-hieu-ve-my-sql-gDVK2Ov2ZLj>