ĐẠI HỌC DUY TÂN

KHOA CÔNG NGHỆ THÔNG TIN

--------------🙖🙐✰🙖🙐-------------



*Đề tài:*

**AIHOTEL PRO- GIẢI PHÁP QUẢN LÝ KHÁCH SẠN THÔNG MINH TÍCH HỢP CHABOT, PREDICTIVE ANALYTICS VÀ THANH TOÁN ĐIỆN TỬ**

**(PRODUCT BACKLOG DOCUMENT)**

**GVHD:** Th.S Hồ Lê Viết Nin

**Thành Viên Nhóm:**

Võ Tuấn Cường - 27212525948

Mai Kiên Thành - 27211202130

Huỳnh Văn Bá Thiện - 27211236697

Trần Quốc Việt - 27214327697

*Đà Nẵng, tháng 09 năm 2025*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **THÔNG TIN DỰ ÁN** | | | | |
| **Dự án viết tắt** | **AIHotel-Pro** | | | |
| **Tên dự án** | AIHotel-Pro Giải pháp quản lý khách sạn thông minh tích hợp chatbot, Predictive Analytics và thanh toán điện tử | | | |
| **Thời gian bắt đầu** | 29/09/2025 | **Thời gian kết thúc** | 15/12/2025 | |
| **Lead Institution** | Khoa Công Nghệ Thông Tin, Trường Đại Học Duy Tân | | | |
| **Giáo viên hướng dẫn** | Hồ Lê Viết Nin  Email: hlvnin88@gmail.com  Phone: 0905455246 | | | |
| **Chủ sở hữu dự án & Chi tiết liên hệ** | Hồ Lê Viết Nin  Email: hlvnin88@gmail.com  Phone: 0905455246 | | | |
| **Đối tác** | Duy Tan University | | | |
| **Quản lý dự án & Scrum Master** | Võ Tuấn Cường | tc091595@gmail.com | | 0357989225 |
| **Thành viên nhóm** | Mai Kiên Thành | kienthanh04072003@gmail.com | | 0386509015 |
| Huỳnh Văn Bá Thiện | thien522003@gmail.com | | 0867798246 |
| Trần Quốc Việt | viet5544g@gmail.com | | 0877171116 |

**TÊN TÀI LIỆU**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề tài liệu** | Product Backlog | | |
| **Tác giả** |  | | |
| **Chức năng** | Thành viên | | |
| **Ngày** | 29/09/2025 | **Tên tệp:** | 03\_[KLTN\_05] Product-Backlog.docx |
| **URL** |  | | |
| **Access** | Khoa CNTT | | |

**LỊCH SỬ BẢN SỬA CHỮA**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Person** | **Date** | **Description** |
| 1.0 |  |  |  |
| 1.1 |  |  |  |

**PHÊ DUYỆT TÀI LIỆU**

Cần có các chữ ký sau để phê duyệt tài liệu này

|  |  |  |  |
| --- | --- | --- | --- |
| **Người hướng dẫn** | Hồ Lê Viết Nin | **Signature:** |  |
| **Date:** |  |
| **Chủ sở hữu** | Hồ Lê Viết Nin | **Signature:** |  |
| **Date:** |  |
| **Đội sản xuất** | Võ Tuấn Cường | **Signature:** |  |
| **Date:** |  |
| **Thành viên** | Mai Kiên Thành | **Signature:** |  |
| **Date:** |  |
| Huỳnh Văn Bá Thiện | **Signature:** |  |
| **Date:** |  |
| Trần Quốc Việt | **Signature:** |  |
| **Date:** |  |

**MỤC LỤC**

[1. GIỚI THIỆU 1](#_Toc211702130)

[1.1 Mục đích 1](#_Toc211702131)

[1.2 Phạm vi 1](#_Toc211702132)

[1.3 Tài liệu tham khảo 1](#_Toc211702133)

[2. PRODUCT BACKLOG 3](#_Toc211702134)

[2.1 Danh sách các product backlog có trong hệ thống 3](#_Toc211702135)

[2.2.1 PB01 - Đăng nhập 7](#_Toc211702136)

[2.2.2 PB02 - Đăng ký 8](#_Toc211702137)

[2.2.3 PB03 - Quản lý thông tin 9](#_Toc211702138)

[2.2.4 PB04 - Quên mật khẩu 10](#_Toc211702139)

[2.2.5 PB05 - Đặt lại mật khẩu 11](#_Toc211702140)

[2.2.6 PB06 - Tìm kiếm 12](#_Toc211702141)

[2.2.7 PB07- Đặt phòng 13](#_Toc211702142)

[2.2.8 PB08 - Sử dụng dịch vụ 14](#_Toc211702143)

[2.2.9 PB09 - Tương tác với chatbot AI 15](#_Toc211702144)

[2.2.10 PB10 - Thanh toán 16](#_Toc211702145)

[2.2.11 PB11- Đánh giá 17](#_Toc211702146)

[2.2.12 PB12 - Quản lý loại phòng 18](#_Toc211702147)

[2.2.13 PB13 - Quản lý dịch vụ 19](#_Toc211702148)

[2.2.14 PB14-Quản lý đặt phòng 20](#_Toc211702149)

[2.2.15 PB15 - Quản lý hóa đơn 21](#_Toc211702150)

[2.2.16 PB16 - Quản lý người dùng 22](#_Toc211702151)

[2.2.17 PB17- Quản lý thống kê 23](#_Toc211702152)

[2.2.18 PB18 - Quản lý chatbot AI 24](#_Toc211702153)

[2.3 Phân tích về Sprint Backlog 25](#_Toc211702154)

[3. Hạn chế 25](#_Toc211702155)

[4. Tóm lược 26](#_Toc211702156)

# 1. GIỚI THIỆU

## 1.1 Mục đích

Tài liệu này trình bày các yêu cầu cấp cao từ góc nhìn của người dùng cuối. Những yêu cầu này sẽ được chia nhỏ thành nhiều nhiệm vụ để phân công cho nhóm phát triển. Ngoài ra, tài liệu cũng bao gồm các tiêu chuẩn chấp nhận, giúp người kiểm tra xây dựng kế hoạch kiểm thử và thực hiện kiểm tra.

## 1.2 Phạm vi

Ứng dụng chạy trên nền tảng Web

Bao gồm các chức năng:

• Liệt kê vai trò người dùng.

• Viết tất cả các yêu cầu của người dùng.

• Liệt kê một số chức năng chính của hệ thống.

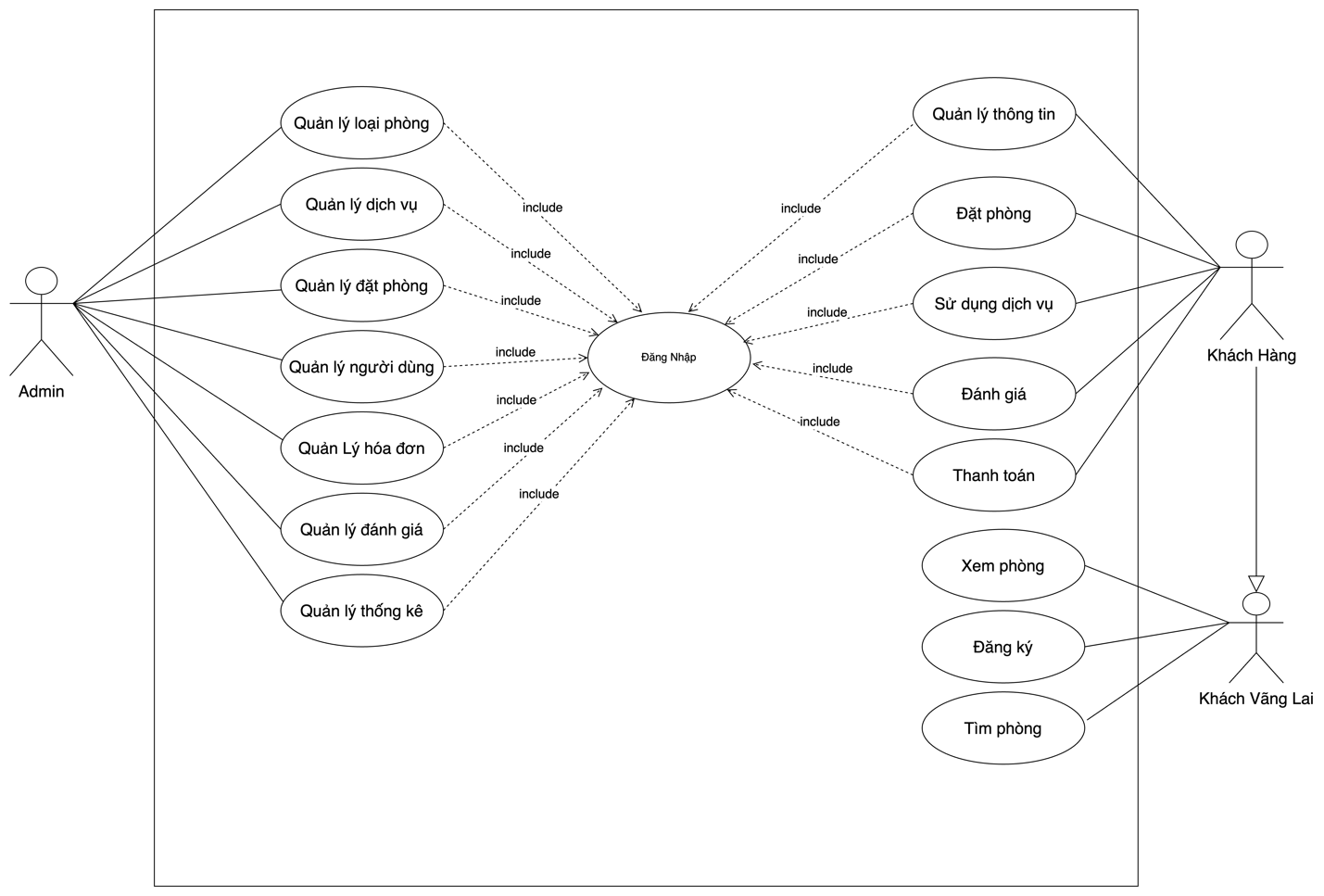
• Mô tả ngắn về tất cả các chức năng mong muốn của sản phẩm.

• Cho độ ưu tiên của mỗi tính năng và chức năng của sản phẩm.

## 1.3 Tài liệu tham khảo

|  |  |  |
| --- | --- | --- |
| **No** | **Reference** | **Note** |
| 1 | http://agilebench.com/blog/the-product-backlog-for-  agile-teams | Cách tạo Product Backlog |
| 2 | http://www.mountaingoatsoftware.com/agile/scrum/pr  oduct-backlog/example/ | Ví dụ về Product Backlog |
| 3 | 01\_[CS445 - 03] Product Proposal.docx | Proposal Document |

**1.4 Biểu đồ Use Case tổng quát**



# 2. PRODUCT BACKLOG

## 2.1 Danh sách các product backlog có trong hệ thống

*Bảng 1: Đặc tả tổng quát các Product Backlog*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Chủ đề** | **Với** | **Mong muốn** | **Nên** | **Sự ưu**  **tiên** |
| PB01 | Đăng nhập | Khách hàng  Admin | Truy cập vào hệ thống | Người dùng có thể đăng nhập vào hệ thống | 1 |
| PB02 | Đăng ký | Khách vãng lai | Thêm tài khoản khoản vào hệ thống | Khách hàng có thể đăng ký tài khoản | 1 |
| PB03 | Thông tin cá nhân | Khách hàng | Cập nhật thông tin cá nhân | Khách hàng có thể xem và cập nhật thông tin của mình | 2 |
| PB04 | Quên mật khẩu | Khách hàng  Admin | Gửi mã về email đã đăng ký tài khoản | Người dùng có thể nhận mã về email đăng ký tài khoản | 2 |
| PB05 | Đặt lại mật khẩu | Khách hàng  Admin | Thay đổi mật khẩu | Người dùng có thể đặt lại mật khẩu của mình | 2 |
| PB06 | Tìm kiếm | Khách vãng lai  Khách hàng  Admin | Tìm kiếm theo từ khóa | Người dùng có thể tìm kiếm theo từ khóa đã nhập | 2 |
| PB07 | Đặt Phòng | Khách hàng | Đặt phòng để lưu trú tại khách sạn | Khách hàng có thể xem thông tin phòng, chọn phòng phù hợp và gửi yêu cầu đặt phòng | 3 |
| PB08 | Sử dụng dịch vụ | Khách hàng | Đặt và sử dụng dịch vụ trong khách sạn | Khách hàng có thể xem danh sách dịch vụ, chọn dịch vụ và gửi yêu cầu sử dụng dịch vụ | 2 |
| PB09 | Tương tác với chatbot AI | Khách hàng | Trao đổi với chatbot AI để được hỗ trợ thông tin về phòng, dịch vụ hoặc các thắc mắc khác | Khách hàng có thể trò chuyện với chatbot để nhận hỗ trợ nhanh chóng, chính xác mà không cần gặp nhân viên | 2 |
| PB10 | Thanh toán | Khách hàng | Khách hàng thành toán hóa đơn | Khách hàng có thể thanh toán thành công hóa đơn đặt phòng | 3 |
| PB11 | Đánh giá | Khách hàng | Gửi đánh giá sau khi sử dụng dịch vụ/phòng | Khách hàng có thể viết đánh giá và gửi phản hồi để cải thiện chất lượng dịch vụ | 2 |
| PB12 | Quản lý loại phòng | Admin | Quản lý danh sách loại phòng trong hệ thống | Admin có thể thêm, sửa, xóa và xem thông tin các loại phòng của khách sạn | 2 |
| PB13 | Quản lý dịch vụ | Admin | Quản lý danh sách dịch vụ trong khách sạn | Admin có thể thêm, sửa, xóa và cập nhật thông tin các dịch vụ phục vụ khách hàng | 2 |
| PB14 | Quản lý đặt phòng | Admin | Theo dõi và xử lý các yêu cầu đặt phòng từ khách hàng | Admin có thể xem, xác nhận, hủy hoặc cập nhật trạng thái các yêu cầu đặt phòng của khách hàng | 2 |
| PB15 | Quản lý hóa đơn | Admin | Quản lý thông tin thanh toán của khách hàng | Admin có thể xem, tìm kiếm, in và xuất hóa đơn thanh toán của khách hàng sau khi sử dụng dịch vụ | 2 |
| PB16 | Quản lý người dùng | Admin | Quản lý danh sách tài khoản người dùng trong hệ thống, bao gồm nhân viên và khách hàng | Admin có thể xem, chỉnh sửa, xóa người dùng và phân quyền (Admin/Nhân viên/Khách hàng) để đảm bảo phân cấp truy cập đúng chức năng | 3 |
| PB17 | Quản lý thống kê | Admin | Xem được thống kê doanh thu, số lượng phòng thuê | Admin có thể xem được thống kê doanh thu, số lượng phòng thuê | 3 |
| PB18 | Quản lý Chatbot AI | Admin | Theo dõi và cập nhật nội dung chatbot, xem thống kê, huấn luyện lại chatbot khi cần | Quản trị viên có thể xem lịch sử hội thoại, thống kê hiệu suất, chỉnh sửa nội dung phản hồi và huấn luyện chatbot để cải thiện chất lượng | 3 |

***\*Giải thích: Độ ưu tiên càng tăng thì càng quan trọng***

### 2.2.1 PB01 - Đăng nhập

|  |  |
| --- | --- |
| **ID** | PB01 |
| **Tiêu đề** | Đăng nhập vào hệ thống |
| **Mô tả** | Người dùng có thể đăng nhập vào hệ thống bằng tài khoản đã đăng ký để sử dụng các chức năng phù hợp với vai trò được phân quyền. |
| **Tiêu chí chấp nhận** | - Có form đăng nhập gồm email và mật khẩu. - Kiểm tra thông tin đăng nhập hợp lệ từ cơ sở dữ liệu. - Thông báo lỗi nếu thông tin không chính xác. - Chuyển hướng vào trang chính khi đăng nhập thành công. - Phân quyền người dùng để hiển thị chức năng phù hợp (Admin, Nhân viên, Khách hàng). |
| **Điều kiện trước** | |  | | --- | |  |   - Người dùng đã có tài khoản hợp lệ trong hệ thống. - Tài khoản phải được cấp quyền truy cập. |
| **Điều kiện sau** | - Người dùng đăng nhập thành công và được điều hướng đúng theo vai trò. - Hệ thống ghi nhận phiên đăng nhập hoặc lưu token xác thực. |
| **Công nghệ sử dụng** | **Frontend**: Vue.js (Form đăng nhập, giao diện) **Backend**: Laravel (Xử lý xác thực) **Database**: MySQL (Lưu thông tin người dùng) **API**: RESTful API (Gửi email/mật khẩu, nhận phản hồi đăng nhập) |
| **Độ phức tạp** | Thấp |

### 2.2.2 PB02 - Đăng ký

|  |  |
| --- | --- |
| **ID** | PB02 |
| **Tiêu đề** | Đăng ký tài khoản |
| **Mô tả** | Người dùng có thể đăng ký tài khoản mới để sử dụng các chức năng của hệ thống như đặt phòng, sử dụng dịch vụ... |
| **Tiêu chí chấp nhận** | - Có form đăng ký gồm: họ tên, email, mật khẩu, xác nhận mật khẩu, số điện thoại. - Kiểm tra hợp lệ định dạng các trường đầu vào (email đúng định dạng, mật khẩu đủ mạnh…). - Kiểm tra trùng lặp email trong hệ thống. - Lưu thông tin người dùng vào cơ sở dữ liệu nếu đăng ký hợp lệ. - Thông báo đăng ký thành công và chuyển hướng đến trang đăng nhập hoặc tự động đăng nhập. |
| **Điều kiện trước** | - Người dùng chưa có tài khoản trên hệ thống. |
| **Điều kiện sau** | - Người dùng có tài khoản hợp lệ trong hệ thống và có thể đăng nhập. |
| **Công nghệ sử dụng** | **Frontend**: Vue.js (Form đăng ký) **Backend**: Laravel (Xử lý lưu dữ liệu, kiểm tra hợp lệ) **Database**: MySQL (Lưu thông tin tài khoản) **API**: RESTful API (Gửi dữ liệu từ form đăng ký đến backend) |
| **Độ phức tạp** | Thấp |

### 2.2.3 PB03 - Quản lý thông tin

|  |  |
| --- | --- |
| **ID** | PB03 |
| **Tiêu đề** | Quản lý thông tin cá nhân |
| **Mô tả** | Người dùng có thể xem và cập nhật thông tin cá nhân trên hệ thống. |
| **Tiêu chí chấp nhận** | - Thiết kế giao diện trang thông tin cá nhân (Hiển thị tên, email, số điện thoại, địa chỉ, ảnh đại diện, v.v.). - Xử lý logic để lấy và hiển thị thông tin cá nhân từ cơ sở dữ liệu. - Tạo tính năng chỉnh sửa thông tin cá nhân cho phép người dùng thay đổi thông tin cơ bản (email, số điện thoại, mật khẩu, địa chỉ, ảnh đại diện). - Cập nhật thông tin vào cơ sở dữ liệu khi người dùng lưu thay đổi. |
| **Điều kiện trước** | |  | | --- | |  |  |  | | --- | | - Thành viên đã đăng nhập vào tài khoản. - Thành viên có thông tin cá nhân đã được lưu trong hệ thống. | |
| **Điều kiện sau** | - Thành viên có thể xem và chỉnh sửa thông tin cá nhân. - Thông tin cá nhân được cập nhật thành công sau khi thay đổi và hệ thống hiển thị thông báo xác nhận. |
| **Công nghệ sử dụng** | **Frontend**: Vue.js (Form đăng ký) **Backend**: Laravel (Xử lý lưu dữ liệu, kiểm tra hợp lệ) **Database**: MySQL (Lưu thông tin tài khoản) **API**: RESTful API (gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Trung Bình |

### 2.2.4 PB04 - Quên mật khẩu

|  |  |
| --- | --- |
| **ID** | PB04 |
| **Tiêu đề** | Quên mật khẩu |
| **Mô tả** | Người dùng có thể khôi phục mật khẩu khi quên thông qua email đã đăng ký. Hệ thống sẽ gửi liên kết đặt lại mật khẩu mới. |
| **Tiêu chí chấp nhận** | - Có form nhập email khôi phục mật khẩu. - Kiểm tra email có tồn tại trong hệ thống. - Gửi email chứa liên kết đặt lại mật khẩu đến email đã đăng ký. - Khi người dùng nhấn vào link, hiển thị form nhập mật khẩu mới và xác nhận. - Cập nhật mật khẩu mới nếu nhập hợp lệ. - Thông báo khi đổi mật khẩu thành công. |
| **Điều kiện trước** | Người dùng đã từng đăng ký tài khoản và còn giữ email đăng ký. |
| **Điều kiện sau** | - Người dùng khôi phục được tài khoản bằng mật khẩu mới. - Có thể đăng nhập lại bình thường bằng mật khẩu mới. |
| **Công nghệ sử dụng** | **Frontend**: Vue.js (Form nhập email, đặt lại mật khẩu) **Backend**: Laravel (Gửi email, xử lý reset mật khẩu) **Database**: MySQL (Cập nhật lại mật khẩu mã hóa) **API**: RESTful API + Laravel Mail (Gửi link khôi phục qua email) |
| **Độ phức tạp** | Trung bình |

### 2.2.5 PB05 - Đặt lại mật khẩu

|  |  |
| --- | --- |
| **ID** | PB05 |
| **Tiêu đề** | Đặt lại mật khẩu |
| **Mô tả** | Người dùng có thể đặt lại mật khẩu mới thông qua liên kết được gửi đến email sau khi yêu cầu quên mật khẩu. |
| **Tiêu chí chấp nhận** | - Liên kết từ email mở ra form đặt lại mật khẩu.  - Form gồm 2 trường: mật khẩu mới và xác nhận mật khẩu.  - Kiểm tra độ mạnh mật khẩu và xác nhận phải khớp.  - Mật khẩu mới được lưu vào hệ thống sau khi mã hóa.  - Thông báo đặt lại thành công và chuyển hướng đến trang đăng nhập. |
| **Điều kiện trước** | Người dùng đã thực hiện bước "quên mật khẩu" và nhận được liên kết đặt lại hợp lệ. |
| **Điều kiện sau** | Mật khẩu người dùng được thay đổi thành công và có thể đăng nhập lại bằng mật khẩu mới. |
| **Công nghệ sử dụng** | **Frontend**: Vue.js (Form đặt lại mật khẩu) **Backend**: Laravel (Xử lý xác nhận token, cập nhật mật khẩu) **Database**: MySQL (Lưu mật khẩu mới đã mã hóa) **API**: RESTful API + Token xác thực |
| **Độ phức tạp** | Trung bình |

### 2.2.6 PB06 - Tìm kiếm

|  |  |
| --- | --- |
| **ID** | PB06 |
| **Tiêu đề** | Tìm kiếm phòng |
| **Mô tả** | Người dùng có thể tìm kiếm các phòng trống trong khách sạn dựa trên các tiêu chí như ngày nhận/trả phòng, loại phòng, số người ở, mức giá, tiện nghi,… |
| **Tiêu chí chấp nhận** | - Có form tìm kiếm với các trường: ngày nhận phòng, ngày trả phòng, loại phòng, số người, giá, tiện nghi. - Kết quả trả về chỉ bao gồm các phòng còn trống, thỏa mãn điều kiện. - Cho phép sắp xếp theo giá, độ phổ biến, loại phòng,… - Cho phép lọc nhanh các tiện ích (có điều hòa, view biển, bồn tắm…). - Nhấn vào từng phòng có thể xem chi tiết. |
| **Điều kiện trước** | - Có danh sách các phòng đang hoạt động và thông tin tình trạng phòng được cập nhật theo thời gian thực. |
| **Điều kiện sau** | - Người dùng xem được danh sách các phòng phù hợp với nhu cầu và có thể chuyển sang bước đặt phòng. |
| **Công nghệ sử dụng** | **Frontend**: Vue.js (Form tìm kiếm, hiển thị kết quả) **Backend**: Laravel (Xử lý logic tìm kiếm, truy vấn điều kiện) **Database**: MySQL (Truy xuất phòng và tình trạng phòng) **API**: RESTful API (Gửi yêu cầu tìm kiếm và nhận kết quả phòng phù hợp) |
| **Độ phức tạp** | Trung bình |

### 2.2.7 PB07- Đặt phòng

|  |  |
| --- | --- |
| **ID** | PB07 |
| **Tiêu đề** | Đặt phòng |
| **Mô tả** | Người dùng có thể đặt phòng trực tuyến bằng cách chọn phòng mong muốn, điền thông tin và xác nhận đặt phòng |
| **Tiêu chí chấp nhận** | - Cho phép chọn phòng từ danh sách phòng trống. - Hiển thị form nhập thông tin: họ tên, số điện thoại, ngày nhận/trả phòng, ghi chú. - Tự động tính tổng tiền dựa trên số ngày và giá phòng. - Kiểm tra phòng có còn trống trong thời gian đặt. - Xác nhận đặt phòng và lưu vào hệ thống. - Gửi thông báo đặt phòng thành công. |
| **Điều kiện trước** | - Người dùng đã đăng nhập. - Có sẵn phòng phù hợp và còn trống trong thời gian người dùng chọn. |
| **Điều kiện sau** | - Hệ thống tạo một yêu cầu đặt phòng và lưu lại trạng thái chờ xác nhận hoặc đã xác nhận. - Gửi thông báo xác nhận đến người dùng (nếu có). |
| **Công nghệ sử dụng** | **Frontend**: Vue.js (Hiển thị thông tin phòng, form đặt phòng) **Backend**: Laravel (Xử lý logic đặt phòng, lưu thông tin) **Database**: MySQL (Lưu thông tin đơn đặt phòng) **API**: RESTful API (Tạo yêu cầu đặt phòng, trả về trạng thái) |
| **Độ phức tạp** | Cao |

### 2.2.8 PB08 - Sử dụng dịch vụ

|  |  |
| --- | --- |
| **ID** | PB08 |
| **Tiêu đề** | Sử dụng dịch vụ |
| **Mô tả** | Khách hàng có thể lựa chọn và yêu cầu sử dụng các dịch vụ đi kèm của khách sạn như ăn uống, giặt ủi, spa,… trong thời gian lưu trú. |
| **Tiêu chí chấp nhận** | - Hiển thị danh sách các dịch vụ đang cung cấp cùng giá. - Cho phép chọn dịch vụ, số lượng và ghi chú nếu có. - Gửi yêu cầu sử dụng dịch vụ cho hệ thống xử lý. - Hệ thống ghi nhận dịch vụ đã chọn và liên kết với đơn đặt phòng. - Cập nhật chi phí dịch vụ vào hóa đơn tổng. |
| **Điều kiện trước** | - Người dùng đã có đơn đặt phòng hợp lệ. - Dịch vụ được khách sạn kích hoạt và còn hiệu lực. |
| **Điều kiện sau** | - Dịch vụ được thêm vào danh sách sử dụng của khách. - Hệ thống cập nhật chi phí vào hóa đơn hoặc bảng thống kê của khách. |
| **Công nghệ sử dụng** | **Frontend**: Vue.js (Danh sách dịch vụ, form yêu cầu dịch vụ) **Backend**: Laravel (Xử lý yêu cầu, cập nhật dữ liệu) **Database**: MySQL (Lưu thông tin dịch vụ đã sử dụng) **API**: RESTful API (Gửi yêu cầu sử dụng dịch vụ, lấy danh sách dịch vụ) |
| **Độ phức tạp** | Trung bình |

### 2.2.9 PB09 - Tương tác với chatbot AI

|  |  |
| --- | --- |
| **ID** | PB09 |
| **Tiêu đề** | Tương tác với chatbot AI |
| **Mô tả** | Khách hàng có thể trò chuyện với chatbot AI để hỏi thông tin về phòng, dịch vụ, khuyến mãi hoặc yêu cầu hỗ trợ trong thời gian lưu trú mà không cần liên hệ trực tiếp với nhân viên. |
| **Tiêu chí chấp nhận** | - Chatbot AI có thể hiểu và phản hồi câu hỏi của khách hàng bằng ngôn ngữ tự nhiên. - Chatbot hiển thị thông tin về phòng, dịch vụ, khuyến mãi khi khách hàng yêu cầu. - Chatbot có thể hướng dẫn khách hàng thực hiện các thao tác như đặt phòng, đặt dịch vụ, hoặc gọi hỗ trợ. - Khi chatbot không xử lý được yêu cầu, hệ thống tự động chuyển tiếp đến nhân viên. - Lưu lại lịch sử hội thoại để nhân viên có thể xem lại và cải thiện phản hồi. - Chatbot phản hồi trong thời gian dưới 5 giây và hoạt động 24/7. |
| **Điều kiện trước** | - Khách hàng đã đăng nhập hệ thống hoặc đang trong trạng thái “đang lưu trú”. - Thiết bị của khách hàng có kết nối internet. |
| **Điều kiện sau** | - Cuộc trò chuyện được lưu trong hệ thống. - Yêu cầu của khách hàng được xử lý hoặc chuyển đến nhân viên phụ trách. |
| **Công nghệ sử dụng** | **Frontend**: Vue.js (Danh sách dịch vụ, form yêu cầu dịch vụ) **Backend**: Laravel (Xử lý yêu cầu, cập nhật dữ liệu) **Database**: MySQL (Lưu thông tin dịch vụ đã sử dụng) **API**: RESTful API (Gửi yêu cầu sử dụng dịch vụ, lấy danh sách dịch vụ) |
| **Độ phức tạp** | Trung bình- Cao |

### 2.2.10 PB10 - Thanh toán

|  |  |
| --- | --- |
| **ID** | PB10 |
| **Tiêu đề** | Thanh toán |
| **Mô tả** | Người dùng có thể thanh toán chi phí đặt phòng và các dịch vụ đã sử dụng thông qua các phương thức như tiền mặt, chuyển khoản, hoặc cổng thanh toán trực tuyến. |
| **Tiêu chí chấp nhận** | - Hiển thị chi tiết hóa đơn gồm: tiền phòng, dịch vụ, thuế, tổng cộng. - Hỗ trợ nhiều hình thức thanh toán: tiền mặt, chuyển khoản, ví điện tử (nếu có). - Khi thanh toán thành công, cập nhật trạng thái hóa đơn và phòng. - Gửi thông báo hoặc hóa đơn điện tử qua email. - Lưu lịch sử thanh toán vào hệ thống. |
| **Điều kiện trước** | - Người dùng đã hoàn tất quá trình lưu trú hoặc đặt phòng. - Có thông tin hóa đơn hợp lệ và tổng tiền cần thanh toán. |
| **Điều kiện sau** | - Hóa đơn được thanh toán và lưu vào hệ thống. - Trạng thái đơn đặt phòng chuyển thành "đã thanh toán" |
| **Công nghệ sử dụng** | **Frontend**: Vue.js (Hiển thị hóa đơn, giao diện thanh toán) **Backend**: Laravel (Xử lý giao dịch, cập nhật trạng thái) **Database**: MySQL (Lưu thông tin hóa đơn, lịch sử thanh toán) **API**: RESTful API + Tích hợp cổng thanh toán (nếu có: Momo, VNPay...) |
| **Độ phức tạp** | Cao |

### 2.2.11 PB11- Đánh giá

|  |  |
| --- | --- |
| **ID** | PB11 |
| **Tiêu đề** | Đánh giá |
| **Mô tả** | Sau khi sử dụng dịch vụ hoặc lưu trú, khách hàng có thể để lại đánh giá về phòng, dịch vụ, hoặc tổng thể trải nghiệm tại khách sạn. |
| **Tiêu chí chấp nhận** | - Chỉ cho phép đánh giá sau khi hoàn tất lưu trú. - Form đánh giá gồm: nội dung đánh giá, số sao (1–5), và lựa chọn phòng/dịch vụ cần đánh giá. - Lưu đánh giá vào hệ thống và hiển thị công khai cho người dùng khác. - Cho phép chỉnh sửa hoặc xóa đánh giá của chính mình. - Quản trị viên có thể duyệt, ẩn hoặc xóa đánh giá nếu cần thiết. |
| **Điều kiện trước** | - Người dùng đã hoàn tất lưu trú hoặc sử dụng dịch vụ. - Người dùng đã đăng nhập vào hệ thống. |
| **Điều kiện sau** | - Đánh giá được lưu và liên kết với đơn đặt phòng, dịch vụ, hoặc phòng cụ thể. - Có thể hiển thị cho các khách hàng khác tham khảo. |
| **Công nghệ sử dụng** | **Frontend**: Vue.js (Form đánh giá, giao diện hiển thị đánh giá) **Backend**: Laravel (Xử lý lưu trữ, kiểm duyệt đánh giá) **Database**: MySQL (Lưu nội dung đánh giá, liên kết người dùng và phòng/dịch vụ) **API**: RESTful API (Gửi/nhận/xóa/sửa đánh giá) |
| **Độ phức tạp** | Trung bình |

### 2.2.12 PB12 - Quản lý loại phòng

|  |  |
| --- | --- |
| **ID** | PB12 |
| **Tiêu đề** | Quản lý loại phòng |
| **Mô tả** | Admin có thể thêm, sửa, xóa và xem danh sách các loại phòng trong hệ thống (ví dụ: Deluxe, Standard, VIP…). |
| **Tiêu chí chấp nhận** | - Hiển thị danh sách loại phòng kèm thông tin: tên loại, mô tả, số người tối đa, giá theo đêm, tiện nghi đi kèm. - Cho phép admin thêm mới loại phòng với đầy đủ thông tin. - Cho phép cập nhật thông tin loại phòng. - Cho phép xóa loại phòng nếu không có phòng nào đang sử dụng loại đó. - Hệ thống kiểm tra hợp lệ dữ liệu khi thêm/sửa. |
| **Điều kiện trước** | - Admin đã đăng nhập và có quyền truy cập trang quản lý. - Hệ thống có giao diện quản lý loại phòng. |
| **Điều kiện sau** | - Danh sách loại phòng được cập nhật đúng theo thao tác của admin. - Các phòng mới tạo có thể liên kết với loại phòng tương ứng. |
| **Công nghệ sử dụng** | **Frontend**: Vue.js (Giao diện danh sách + form thêm/sửa loại phòng) **Backend**: Laravel (Xử lý CRUD loại phòng) **Database**: MySQL (Lưu thông tin loại phòng) **API**: RESTful API (Thêm/sửa/xóa/lấy danh sách loại phòng |
| **Độ phức tạp** | Trung bình |

### 2.2.13 PB13 - Quản lý dịch vụ

|  |  |
| --- | --- |
| **ID** | PB13 |
| **Tiêu đề** | Quản lý dịch vụ |
| **Mô tả** | Admin có thể thêm, sửa, xóa và xem danh sách các dịch vụ mà khách sạn cung cấp như giặt ủi, ăn uống, spa, thuê xe,… |
| **Tiêu chí chấp nhận** | - Hiển thị danh sách dịch vụ với các thông tin: tên dịch vụ, mô tả, giá, trạng thái hoạt động. - Cho phép thêm mới dịch vụ với đầy đủ thông tin. - Cho phép chỉnh sửa thông tin dịch vụ. - Cho phép xóa dịch vụ nếu chưa có khách hàng sử dụng. - Kiểm tra hợp lệ dữ liệu trước khi thêm/sửa. |
| **Điều kiện trước** | - Admin đã đăng nhập và có quyền truy cập trang quản lý dịch vụ. |
| **Điều kiện sau** | - Danh sách dịch vụ được cập nhật tương ứng trong hệ thống. - Khách hàng có thể xem và lựa chọn các dịch vụ đã được cập nhật |
| **Công nghệ sử dụng** | **Frontend**: Vue.js (Giao diện danh sách + form thêm/sửa dịch vụ) **Backend**: Laravel (Xử lý CRUD dịch vụ) **Database**: MySQL (Lưu thông tin dịch vụ) **API**: RESTful API (Thêm/sửa/xóa/lấy danh sách dịch vụ) |
| **Độ phức tạp** | Trung bình |

### 2.2.14 PB14-Quản lý đặt phòng

|  |  |
| --- | --- |
| **ID** | PB14 |
| **Tiêu đề** | Quản lý đặt phòng |
| **Mô tả** | Admin có thể xem, xác nhận, hủy, chỉnh sửa và quản lý toàn bộ các đơn đặt phòng của khách hàng trong hệ thống. |
| **Tiêu chí chấp nhận** | - Hiển thị danh sách các đơn đặt phòng: mã đơn, tên khách hàng, ngày nhận/trả, trạng thái, tổng tiền. - Cho phép admin xác nhận hoặc hủy đơn đặt phòng. - Cho phép chỉnh sửa thông tin đặt phòng nếu cần (ngày, loại phòng, ghi chú). - Có bộ lọc theo trạng thái (đã đặt, đang xử lý, đã xác nhận, đã hủy). - Cập nhật trạng thái phòng theo tình trạng đơn. - Hệ thống gửi thông báo cho khách khi đơn được xử lý. |
| **Điều kiện trước** | - Admin đã đăng nhập và có quyền truy cập quản lý đặt phòng. - Có đơn đặt phòng trong hệ thống. |
| **Điều kiện sau** | - Đơn đặt phòng được cập nhật đúng theo thao tác quản lý. - Trạng thái phòng, hóa đơn và thông báo cũng được cập nhật tương ứng. |
| **Công nghệ sử dụng** | **Frontend**: Vue.js (Danh sách đơn + form chi tiết + chức năng xác nhận/hủy) **Backend**: Laravel (Xử lý logic đặt phòng, cập nhật trạng thái) **Database**: MySQL (Lưu thông tin đặt phòng và trạng thái liên quan) **API**: RESTful API (Lấy danh sách đơn, cập nhật trạng thái, sửa đơn) |
| **Độ phức tạp** | Trung bình |

### 2.2.15 PB15 - Quản lý hóa đơn

|  |  |
| --- | --- |
| **ID** | PB15 |
| **Tiêu đề** | Quản lý hóa đơn |
| **Mô tả** | Admin có thể xem, tìm kiếm, chỉnh sửa, in và quản lý toàn bộ hóa đơn thanh toán của khách hàng trong hệ thống. |
| **Tiêu chí chấp nhận** | - Hiển thị danh sách hóa đơn gồm: mã hóa đơn, khách hàng, ngày lập, tổng tiền, trạng thái thanh toán. - Cho phép tìm kiếm và lọc hóa đơn theo thời gian, khách hàng, trạng thái. - Cho phép xem chi tiết hóa đơn (tiền phòng, dịch vụ, thuế…). - Cho phép cập nhật trạng thái thanh toán thủ công nếu thanh toán offline. - Cho phép in hóa đơn hoặc xuất PDF. - Hệ thống liên kết hóa đơn với đặt phòng và khách hàng. |
| **Điều kiện trước** | - Admin đã đăng nhập và có quyền quản lý hóa đơn. - Có hóa đơn trong hệ thống. |
| **Điều kiện sau** | - Thông tin hóa đơn được cập nhật, theo dõi chính xác. - Hệ thống ghi nhận thanh toán đúng trạng thái và liên kết dữ liệu liên quan. |
| **Công nghệ sử dụng** | **Frontend**: Vue.js (Bảng hóa đơn, chi tiết hóa đơn, form cập nhật) **Backend**: Laravel (Xử lý truy vấn, cập nhật hóa đơn, xuất file) **Database**: MySQL (Lưu thông tin hóa đơn, trạng thái, liên kết khách hàng/đặt phòng) **API**: RESTful API (Lấy danh sách, cập nhật, in hóa đơn) |
| **Độ phức tạp** | Trung bình |

### 2.2.16 PB16 - Quản lý người dùng

|  |  |
| --- | --- |
| **ID** | PB16 |
| **Tiêu đề** | Quản lý người dùng (nhân viên, khách hàng, phân quyền) |
| **Mô tả** | Admin có thể quản lý thông tin người dùng bao gồm nhân viên và khách hàng, đồng thời gán quyền truy cập tương ứng cho từng loại tài khoản. |
| **Tiêu chí chấp nhận** | - Hiển thị danh sách người dùng với thông tin: tên, email, số điện thoại, vai trò (admin, nhân viên, khách hàng), trạng thái hoạt động. - Cho phép tạo mới tài khoản nhân viên và phân quyền truy cập. - Cho phép chỉnh sửa hoặc vô hiệu hóa tài khoản. - Tìm kiếm, lọc người dùng theo vai trò hoặc trạng thái. - Quản lý quyền theo vai trò: giới hạn quyền của nhân viên và khách hàng. - Đảm bảo bảo mật thông tin và không hiển thị mật khẩu người dùng. |
| **Điều kiện trước** | - Admin đã đăng nhập và có quyền quản trị người dùng. - Các vai trò và quyền đã được định nghĩa trong hệ thống. |
| **Điều kiện sau** | - Danh sách người dùng được cập nhật chính xác. - Quyền truy cập hệ thống được áp dụng đúng theo phân quyền đã gán. |
| **Công nghệ sử dụng** | **Frontend**: Vue.js (Bảng danh sách người dùng, form chỉnh sửa, form phân quyền) **Backend**: Laravel (Xử lý CRUD người dùng, xác thực, phân quyền) **Database**: MySQL (Lưu thông tin người dùng, vai trò, quyền) **API**: RESTful API (Quản lý người dùng và quyền) |
| **Độ phức tạp** | Cao |

### 2.2.17 PB17- Quản lý thống kê

|  |  |
| --- | --- |
| **ID** | PB17 |
| **Tiêu đề** | Quản lý thống kê (doanh thu, số phòng, số phòng đã thuê) |
| **Mô tả** | Admin có thể xem các báo cáo thống kê về hoạt động kinh doanh của khách sạn như doanh thu theo ngày/tháng/năm, tổng số lượng phòng, và số phòng đã được thuê. |
| **Tiêu chí chấp nhận** | - Hiển thị biểu đồ thống kê doanh thu theo thời gian (ngày/tháng/năm). - Hiển thị tổng số lượng phòng đang hoạt động. - Hiển thị số lượng phòng đã được thuê theo thời gian. - Có bộ lọc theo khoảng thời gian để thống kê chi tiết. - Dữ liệu được cập nhật theo thời gian thực. |
| **Điều kiện trước** | - Admin đã đăng nhập và có quyền xem thống kê. - Hệ thống có dữ liệu đặt phòng, hóa đơn và phòng. |
| **Điều kiện sau** | - Dữ liệu thống kê được hiển thị chính xác. - Admin có thể sử dụng dữ liệu để đưa ra quyết định quản lý. |
| **Công nghệ sử dụng** | **Frontend**: Vue.js (Giao diện hiển thị biểu đồ, bảng dữ liệu, bộ lọc thống kê) **Backend**: Laravel (Xử lý truy vấn thống kê, tổng hợp dữ liệu) **Database**: MySQL (Dữ liệu từ bảng hóa đơn, đặt phòng, phòng) **Thư viện biểu đồ**: Chart.js hoặc ApexCharts **API**: RESTful API (Truy vấn thống kê theo thời gian) |
| **Độ phức tạp** | Cao |

### 2.2.18 PB18 - Quản lý chatbot AI

|  |  |
| --- | --- |
| **ID** | PB18 |
| **Tiêu đề** | Quản lý Chatbot AI |
| **Mô tả** | Quản trị viên có thể theo dõi hoạt động của chatbot, xem lịch sử hội thoại, thống kê hiệu suất, chỉnh sửa nội dung phản hồi và huấn luyện lại chatbot để nâng cao chất lượng hỗ trợ khách hàng. |
| **Tiêu chí chấp nhận** | - Quản trị viên có thể đăng nhập vào giao diện quản lý chatbot. - Xem được danh sách và nội dung các cuộc trò chuyện giữa khách hàng và chatbot. - Chỉnh sửa hoặc thêm mới các kịch bản phản hồi của chatbot. - Có chức năng huấn luyện lại chatbot để cải thiện độ chính xác. - Mọi thay đổi được ghi vào nhật ký hệ thống (log) để theo dõi |
| **Điều kiện trước** | - Quản trị viên đã đăng nhập hệ thống với quyền “Admin”. - Chatbot đã được triển khai và đang hoạt động. |
| **Điều kiện sau** | - Cập nhật mới được áp dụng cho chatbot. - Thống kê, log và lịch sử thay đổi được lưu lại trong hệ thống. |
| **Công nghệ sử dụng** | **Frontend:** Vue.js (Giao diện quản lý, bảng thống kê, form chỉnh sửa phản hồi) **Backend:** Laravel (Xử lý quản lý dữ liệu, cập nhật nội dung chatbot) **Database:** MySQL (Lưu hội thoại, log thay đổi, thống kê) **AI Service:** Dialogflow / OpenAI API (Huấn luyện và cập nhật mô hình chatbot) **API:** RESTful API |
| **Độ phức tạp** | Cao |

## 2.3 Phân tích về Sprint Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprint** | **Function** | **Started Date** | **Due Date** |
| Sprint 1 | Develop | 05/10/2025 | 10/11/2025 |
| Sprint 2 | Develop | 11/11/2025 | 11/12/2025 |

## 3. Hạn chế

|  |  |
| --- | --- |
| **Hạn chế** | **Tình trạng** |
| **Thời gian** | Thời gian để hoàn thành dự án |
| **Nhân lực** | Toàn bộ nhóm, tổng cộng 4 người làm việc cùng nhau để hoàn thành dự án. |
| **Các yêu cầu** | Dựa trên các yêu cầu của Product owner. |

## 4. Tóm lược

|  |  |  |
| --- | --- | --- |
| **Name** | **Mô tả** | **Role** |
| **Product Owner** | Người đưa ra yêu cầu. | Cung cấp thông tin để phát triển hệ thống. Ra quyết định nghiệm thu và triển khai dự án, làm unit test. |
| **Scrum Master** | Đây là bên liên quan lãnh đạo, quản lý Nhóm phát triển hệ thống. | Kiểm soát, quản lý, giám sát, đảm bảo dự án hoàn thành đúng thời hạn, ngân sách, đúng kế hoạch và yêu cầu. |
| **Requirement**  **Analyzer** | Đây là một bên liên quan làm việc với các Nhà phân tích để truyền đạt chính xác các yêu cầu hoặc nhu cầu | Chỉ định chi tiết của một hoặc nhiều phần chức năng của hệ thống bằng cách mô tả một hoặc các khía cạnh của yêu cầu. |
|  | thành các yêu cầu được sử dụng cho thiết kế. | Điều này sẽ bao gồm chức năng và phi chức năng. |
| **Software Architect** | Đây là một bên liên quan nắm vai trò chính, dẫn dắt sự phát triển hệ thống. | Chịu trách nhiệm về kiến trúc phần mềm, bao gồm các quyết định kỹ thuật quan trọng hạn chế thiết kế tổng thể cho dự án. |
| **Coder** | Đây là bộ phận lập trình phần mềm. | Thực hiện dự án. |