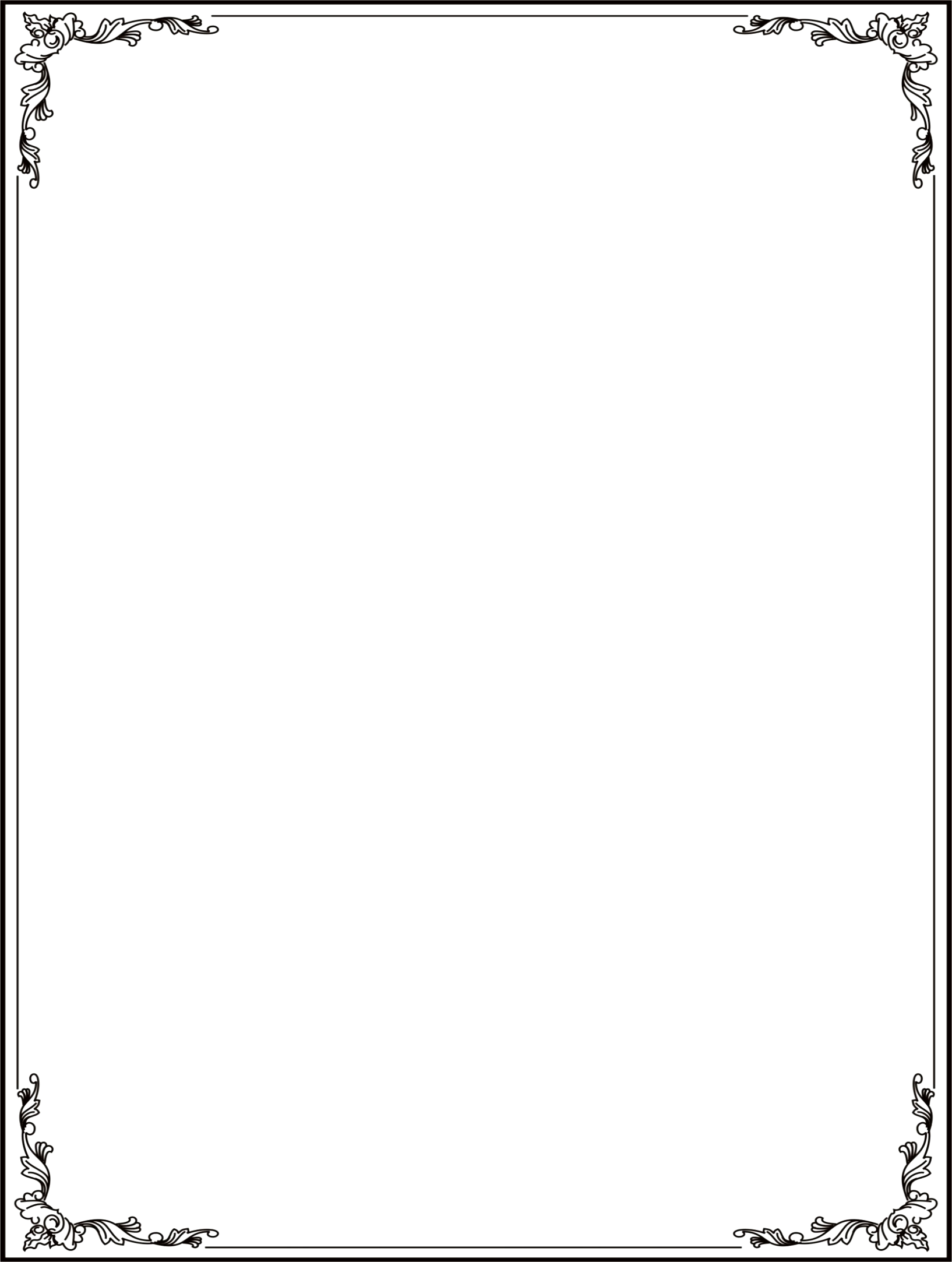
---------------TRƯỜNG ĐẠI HỌC XÂY DỰNG HÀ NỘI

KHOA CÔNG NGHỆ THÔNG TIN

\*\*\*\*\*

Icon

Description automatically generated

**ĐỒ ÁN THIẾT KẾ KIẾN TRÚC PHẦN MỀM**

**Đề tài**

**HỆ THỐNG QUẢN LÝ GARAGE VÀ ĐẶT LỊCH ONLINE**

Nhóm sinh viên thực hiện: Nhóm 46

Danh sách sinh viên: **DƯƠNG TẤT HƯNG <0192266>**

**ĐẶNG MINH KHÔI <0193166>**

Lớp học phần: 66PM

Giảng viên hướng dẫn: **Phạm Hữu Tùng**

Hà Nội 05/2024

**Mục lục**

[Chương 1: MÔ TẢ BÀI TOÁN 4](#_Toc16832)

[1.1. Lý do chọn đề tài 4](#_Toc18623)

[1.2. Mục tiêu của đồ án 4](#_Toc20796)

[1.3. Phạm vi và giới hạn 4](#_Toc12234)

[1.4. Phương pháp thực hiện 5](#_Toc31427)

[Chương 2. Phân tích hệ thống 6](#_Toc14812)

[2.1. Mô tả bài toán 6](#_Toc24758)

[2.2. Yêu cầu chức năng 6](#_Toc13093)

[2.2.1. Chức năng dành cho khách hàng 6](#_Toc19601)

[2.2.2. Chức năng dành cho quản trị viên 6](#_Toc23832)

[2.3. Yêu cầu phi chức năng 6](#_Toc10320)

[2.4. Use Case Diagram tổng quát 7](#_Toc22189)

[2.5. Mô tả các Use Case chính 8](#_Toc18491)

[2.5.1 Usecase Booking Repair Appointment 8](#_Toc14223)

[2.5.2 Usecase Manage My Repair Appointment 10](#_Toc12232)

[2.5.3 Usecase Manage Appointment 14](#_Toc32241)

[2.6 Biểu đồ quy trình đặt lịch 20](#_Toc4775)

[Chương 3: Biểu đồ phân tích hệ thống 22](#_Toc6980)

[3.1. Biểu đồ tuần tự 22](#_Toc6227)

[3.2. Thiết kế lớp (Class Diagram) 29](#_Toc2623)

[3.3. Thiết kế cơ sở dữ liệu 29](#_Toc22822)

[3.4. Thiết kế giao diện người dùng 31](#_Toc7841)

[Chương 4. Cài đặt và triển khai hệ thống 34](#_Toc5900)

[4.1. Môi trường triển khai 34](#_Toc29926)

[4.2. Cài đặt hệ thống 34](#_Toc24826)

[4.2.1 Yêu cầu hệ thống 34](#_Toc15112)

[4.2.2 Các bước cài đặt chương trình 34](#_Toc1136)

[Chương 5. Kết quả và đánh giá 36](#_Toc18992)

[5.1. Kết quả thử nghiệm hệ thống 36](#_Toc22774)

[5.2. Đánh giá hiệu quả hệ thống 36](#_Toc31426)

**Mục lục hình ảnh**

[Hình 1 : UseCase tổng quan 7](#_Toc21701)

[Hình 2 : Quy trình nghiệp vụ 21](#_Toc401)

[Hình 3 : Biểu đồ trình tự Booking Repair Appointment 23](#_Toc9307)

[Hình 4 : Biểu đồ trình tự Manage Repare Appointment 23](#_Toc2634)

[Hình 5 : Biểu đồ trình tự View Repair Appointment 23](#_Toc22499)

[Hình 6 : Biểu đồ trình tự Update Repair Appointment 24](#_Toc21633)

[Hình 7 : Biểu đồ trình tự Cancel Repair Appointment 25](#_Toc5801)

[Hình 8 : Biểu đồ trình tự Create Repair Appointment ( Admin) 25](#_Toc16385)

[Hình 9 : Biểu đồ trình tự Search Repair Appointment (Admin) 26](#_Toc11756)

[Hình 10 : Biểu đồ trình tự Update Repair Appointment ( Admin) 27](#_Toc27673)

[Hình 11 : Biểu đồ trình tự Delete Repair Appointment (Admin) 28](#_Toc24719)

[Hình 12 : Biểu đồ lớp 29](#_Toc9456)

[Hình 13 : Biểu đồ ERD 30](#_Toc24337)

[Hình 14 Giao diện đặt lịch 31](#_Toc15129)

[Hình 15 : Giao diện người dùng nhập thông tin đặt lịch 32](#_Toc31679)

[Hình 16 : Giao diện theo dõi lịch hẹn phía người dùng 33](#_Toc31079)

[Hình 17 : Giao diện admin 33](#_Toc28456)

# 

# Chương 1: MÔ TẢ BÀI TOÁN

## 1.1. Lý do chọn đề tài

Trong bối cảnh nhu cầu sửa chữa và bảo dưỡng ô tô ngày càng tăng cao, việc quản lý lịch hẹn thủ công tại các gara đang trở nên kém hiệu quả và gây nhiều khó khăn cho cả chủ gara và khách hàng. Chủ gara thường gặp khó khăn trong việc sắp xếp lịch hẹn, quản lý doanh thu, cũng như kiểm soát tình trạng phương tiện và dịch vụ. Đồng thời, khách hàng cũng gặp nhiều bất tiện trong việc tìm kiếm gara uy tín, đặt lịch sửa chữa nhanh chóng, và theo dõi tiến độ hoặc thanh toán dịch vụ một cách thuận tiện. Do đó, việc xây dựng một hệ thống quản lý gara và đặt lịch sửa chữa xe trực tuyến trở nên cần thiết nhằm giải quyết những vấn đề này, giúp nâng cao hiệu quả quản lý và trải nghiệm của người dùng.

## 1.2. Mục tiêu của đồ án

* Tối ưu hóa quy trình đặt lịch sửa chữa, giúp khách hàng dễ dàng tìm kiếm dịch vụ, đặt lịch, thanh toán, và quản lý lịch hẹn một cách thuận tiện.
* Hỗ trợ chủ gara trong việc quản lý hiệu quả, bao gồm theo dõi tình trạng phương tiện, và lịch hẹn sửa chữa.
* Cung cấp một hệ thống minh bạch, tiện lợi và dễ sử dụng, góp phần hiện đại hóa quy trình quản lý gara truyền thống.

## 1.3. Phạm vi và giới hạn

* **Phạm vi hệ thống:**
  + Hỗ trợ khách hàng đặt lịch sửa chữa một cách nhanh chóng, tiện lợi thông qua nền tảng web.
  + Hệ thống triển khai tập trung cho một hoặc một số gara tại một khu vực địa lý cụ thể, ưu tiên các dịch vụ phổ biến như bảo dưỡng định kỳ, thay dầu, sửa chữa gầm máy, điện - điều hòa.
  + Bao gồm các chức năng chính như quản lý lịch hẹn, đặt lịch hẹn.
* **Giới hạn hệ thống:**
  + Hiện tại hệ thống chưa phát triển ứng dụng trên nền tảng di động (mobile app).
  + Không bao gồm dịch vụ nâng cấp, độ xe, hoặc các dịch vụ đặc biệt phức tạp..
  + Hỗ trợ thanh toán trực tuyến hoặc tại gara bằng tiền mặt.

## 1.4. Phương pháp thực hiện

 **Phân tích yêu cầu**: Thu thập, phân tích và xác định các yêu cầu chức năng và phi chức năng của hệ thống dựa trên nhu cầu thực tế của chủ sân và khách hàng.

 **Thiết kế hệ thống**: Sử dụng mô hình kiến trúc **MVC**  gồm các tầng:

* **Model:** Đại diện cho dữ liệu và logic nghiệp vụ của ứng dụng.
* **View:** Đại diện cho giao diện người dùng, hiển thị dữ liệu từ Model.
* **Controller**  Làm trung gian giữa Model và View, xử lý các sự kiện và yêu cầu của người dùng, sau đó cập nhật dữ liệu trong Model và hiển thị lại View.

 **Phát triển frontend**: Sử dụng **HTML, CSS** để xây dựng giao diện người dùng hiện đại, trực quan.

 **Phát triển backend**: Dùng **Laravel** làm framework PHP để phát triển hệ thống API, xử lý nghiệp vụ, phân quyền người dùng, và đảm bảo an toàn bảo mật hệ thống.

 **Quản lý công việc**:

* Sử dụng **Trello** để lập kế hoạch, phân công nhiệm vụ, theo dõi tiến độ phát triển theo mô hình Agile (Scrum).
* Các công việc được chia thành sprint, giúp dễ theo dõi và điều phối nhóm phát triển.



**Kiểm soát mã nguồn**:

* Dùng **GitHub** để quản lý source code, thực hiện version control, tạo pull request, quản lý nhánh (branch) cho từng tính năng.
* Link GitHub: https://github.com/Po1211/DoAnTKKTPM\_GarageWebsite

# Chương 2. Phân tích hệ thống

## 2.1. Mô tả bài toán

Bài toán đặt ra là xây dựng một hệ thống quản lý sân bóng và đặt lịch đá bóng online nhằm thay thế phương thức quản lý thủ công truyền thống đang tồn tại nhiều hạn chế và bất tiện.

Cụ thể, hệ thống cần giải quyết các vấn đề sau:

* **Quản lý lịch lịch hẹn của tôi:** Cho phép khách hàng dễ dàng theo dõi và đặt sân lịch hẹn theo ngày và khung giờ cụ thể, tránh xung đột khi nhiều khách hàng đặt sân cùng lúc.
* **Quản lý lịch hẹn:** Cho phép admin dễ dàng theo dõi lịch hẹn và tình trạng mỗi lịch hẹn, dễ dàng cập nhật khi cần thiết.

Mục tiêu của bài toán là xây dựng một hệ thống giúp số hóa, tối ưu hóa và tự động hóa quy trình quản lý garage và đặt lịch hẹn, nâng cao hiệu quả quản lý cũng như trải nghiệm người dùng.

## 2.2. Yêu cầu chức năng

### 2.2.1. Chức năng dành cho khách hàng

* **Đặt sân trực tuyến:** Chọn ngày, giờ đặt sân; nhận xác nhận và theo dõi lịch sử đặt sân.
* **Quản lý lịch hẹn của tôi:** Xem, hủy, thay đổi lịch đặt trước hạn định; lưu lịch sử đặt sân để tra cứu.

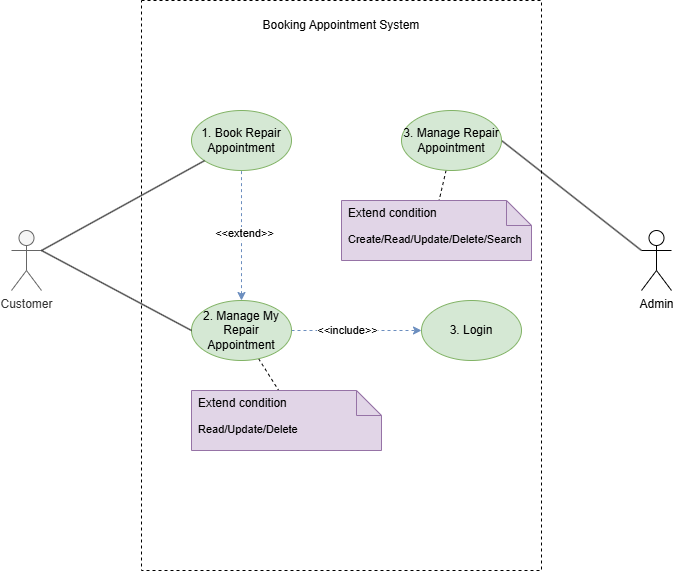
### 2.2.2. Chức năng dành cho quản trị viên

* **Quản lý lịch đặt sân:** Theo dõi lịch đặt theo ngày, tuần, tháng để điều phối và tránh trùng lịch.

## 2.3. Yêu cầu phi chức năng

* **Tính dễ sử dụng:** Giao diện thân thiện, dễ thao tác với đa dạng nhóm người dùng.
* **Hiệu năng:** Phản hồi các thao tác tìm kiếm, đặt sân nhanh (dưới 2 giây).
* **Bảo mật:** Bảo vệ dữ liệu cá nhân, thanh toán và ngăn chặn truy cập trái phép.
* **Khả năng mở rộng:** Hỗ trợ mở rộng cho nhiều sân và khu vực trong tương lai.
* **Tính ổn định:** Hoạt động liên tục với tỷ lệ downtime thấp; bảo toàn dữ liệu.
* **Tương thích trình duyệt:** Frontend React hỗ trợ các trình duyệt phổ biến như Chrome, Firefox, Edge.

## 2.4. Use Case Diagram tổng quát



Hình 1: UseCase tổng quan

## 2.5. Mô tả các Use Case chính

### 2.5.1 Usecase Booking Repair Appointment

|  |  |
| --- | --- |
| Use Case ID | UC-1 |
| Use Case Name | Booking Repair Appointment |
| Description | Là khách hàng, tôi muốn đặt lịch hẹn sửa chữa xe bao gồm việc điền thông tin xe, chọn thời gian và xác nhận lịch hẹn, để tiện lợi sửa chữa xe của mình vào thời gian phù hợp. |
| Actor(s) | Customer |
| Trigger | Khách hàng truy cập vào hệ thống đặt lịch để lên lịch sửa chữa qua website |
| Pre-Condition(s): | * Khách hàng có kết nối internet và có thể truy cập hệ thống. * Hệ thống đang hoạt động bình thường. * Khách hàng có thể là người dùng đã đăng nhập (thông tin cá nhân, xe có sẵn) hoặc người dùng mới cần nhập thông tin thủ công. |
| Post-Condition(s): | * Lịch hẹn sửa chữa được lưu trữ trong hệ thống. * Khách hàng nhận được thông báo xác nhận lịch hẹn qua email. |
| Basic Flow | 1. Khách hàng truy cập giao diện đặt lịch sửa chữa trên website. 2. Hệ thống yêu cầu khách hàng điền thông tin xe .   a. Nếu khách đã đăng nhập, hệ thống tự động hiển thị thông tin đã lưu.  b. Nếu khách chưa đăng nhập, hệ thống yêu cầu khách nhập thông tin thủ công.  3. Hệ thống hiển thị danh sách dịch vụ phù hợp với thông tin xe đã nhập.  4. Khách hàng chọn dịch vụ cần thực hiện.  5. Hệ thống hiển thị các khung giờ trống tương ứng với dịch vụ đã chọn.  6. Khách hàng chọn thời gian sửa chữa phù hợp.  7. Hệ thống yêu cầu xác nhận thông tin đặt lịch.  8. Khách hàng xác nhận lịch hẹn.  9. Hệ thống lưu lịch hẹn và gửi xác nhận qua email. |
| Alternative Flow | 2a. Nếu khách hàng không điền đủ hoặc sai thông tin xe:  2a1. Hệ thống yêu cầu khách hàng nhập lại đúng và đầy đủ.  2a2. Khách hàng nhập lại, tiếp tục bước 3. |
| Exception Flow | 9a. Nếu gửi xác nhận không thành công:  9a1. Hệ thống thông báo lỗi gửi xác nhận.  9a2. Use Case không thực hiện thành công. |

### 2.5.2 Usecase Manage My Repair Appointment

|  |  |
| --- | --- |
| Use Case ID | UC-2 |
| Use Case Name | Manage My Repair Appointment |
| Description | Là khách hàng, tôi muốn quản lý các lịch hẹn sửa chữa xe của mình, bao gồm việc tạo mới, xem, cập nhật và hủy lịch hẹn, để tôi có thể kiểm soát và điều chỉnh kế hoạch sửa chữa một cách dễ dàng. |
| Actor(s) | Customer |
| Trigger | Người dùng chọn "Quản lý lịch hẹn sửa chữa" từ giao diện hệ thống. |
| Pre-Condition(s): | * Người dùng đã đăng nhập vào hệ thống. |
| Post-Condition(s): | * Hành động tương ứng (tạo, xem, cập nhật, xóa) được thực hiện và phản ánh trong hệ thống. |
| Basic Flow | 1. Người dùng chọn “Quản lý lịch hẹn”. 2. Hệ thống hiển thị danh sách lịch hẹn hiện tại (nếu có). 3. Người dùng chọn một trong các thao tác: 4. Tạo lịch hẹn mới. 5. Xem chi tiết lịch hẹn. 6. Cập nhật thông tin lịch hẹn. 7. Hủy lịch hẹn. 8. Hệ thống điều hướng đến luồng xử lý tương ứng với thao tác đã chọn. |
| Alternative Flow | 2a. Nếu người dùng không có lịch sử sửa chữa:  2a1. Hệ thống hiển thị thông báo: “Bạn chưa có lịch sử sửa chữa nào.”  2a2. Use Case kết thúc. |
| Exception Flow |  |

|  |  |
| --- | --- |
| Use Case ID | UC-2.1 |
| Use Case Name | View Repair Appointment Details |
| Description | Là khách hàng, tôi muốn xem chi tiết lịch hẹn sửa chữa xe của mình, để kiểm tra lại thông tin. |
| Actor(s) | Customer |
| Trigger | Người dùng chọn tùy chọn "Xem lịch sử sửa chữa" trong hệ thống. |
| Pre-Condition(s): | * Người dùng đã đăng nhập vào hệ thống. * Hệ thống có thể truy vấn dữ liệu lịch sử sửa chữa (nếu có) của người dùng từ cơ sở dữ liệu. |
| Post-Condition(s): | * Thông tin chi tiết của lịch hẹn sửa chữa được hiển thị cho người dùng. |
| Basic Flow | 1. Người dùng chọn "Xem lịch sử sửa chữa" từ giao diện hệ thống.  2. Hệ thống truy vấn cơ sở dữ liệu để lấy thông tin lịch sử sửa chữa.  3. Nếu có lịch hẹn, hệ thống hiển thị chi tiết lịch hẹn sửa chữa cho người dùng. |
| Alternative Flow | 3a. Nếu người dùng không có lịch sử sửa chữa:  3a1. Hệ thống hiển thị thông báo: “Bạn chưa có lịch sử sửa chữa nào.”  3a2. Use Case kết thúc. |
| Exception Flow |  |

|  |  |
| --- | --- |
| Use Case ID | UC-2.2 |
| Use Case Name | Update Repair Appointment |
| Description | Là khách hàng, tôi muốn cập nhật lịch hẹn sửa chữa xe của mình, để lịch hẹn luôn chính xác và phù hợp với nhu cầu của tôi. |
| Actor(s) | Customer |
| Trigger | * Người dùng chọn tùy chọn "Cập nhật lịch hẹn" trong hệ thống. |
| Pre-Condition(s): | * Người dùng đã đăng nhập vào hệ thống. * Lịch hẹn sửa chữa đã được tạo trước đó. |
| Post-Condition(s): | * Lịch hẹn sửa chữa được cập nhật với thông tin mới trong cơ sở dữ liệu. * Người dùng nhận được xác nhận thông tin đã cập nhật thông qua email. |
| Basic Flow | 1. Người dùng chọn "Cập nhật lịch hẹn". 2. Hệ thống hiển thị thông tin cũ của lịch hẹn. 3. Người dùng thay đổi thông tin cần thiết 4. Hệ thống cập nhật thông tin vào cơ sở dữ liệu. 5. Hệ thống gửi thông báo xác nhận cập nhật lịch hẹn email. |
| Alternative Flow | 3a. Nếu khách hàng không điền đủ hoặc sai thông tin xe:  3a1. Hệ thống yêu cầu khách hàng nhập lại đúng và đầy đủ.  3a2. Khách hàng nhập lại, tiếp tục bước 4. |
| Exception Flow | 5a. Nếu gửi xác nhận không thành công:  5a1. Hệ thống thông báo lỗi gửi xác nhận.  5a2. Use Case không thực hiện thành công. |

|  |  |
| --- | --- |
| Use Case ID | UC-2.3 |
| Use Case Name | Delete Repair Appointment |
| Description | Là khách hàng, tôi muốn hủy lịch hẹn sửa chữa xe của mình, để thông tin của tôi được cập nhật chính xác. |
| Actor(s) | Customer |
| Trigger | Người dùng chọn tùy chọn "Xóa lịch hẹn" trong hệ thống. |
| Pre-Condition(s): | * Người dùng đã đăng nhập vào hệ thống. * Lịch hẹn sửa chữa đã được tạo trước đó. |
| Post-Condition(s): | * Lịch hẹn sửa chữa được cập nhật trạng trạng thái đã hủy trong cơ sở dữ liệu và hiển thị trên hệ thống * Người dùng nhận được thông báo xác nhận hủy lịch hẹn thành công thông qua email. |
| Basic Flow | 1. Người dùng chọn "Hủy lịch hẹn". 2. Hệ thống yêu cầu người dùng xác nhận hủy. 3. Người dùng xác nhận hủy lịch hẹn. 4. Hệ thống cập nhật trạng thái hủy trong cơ sở dữ liệu. 5. Hệ thống gửi thông báo xác nhận hủy lịch hẹn thành công về email. |
| Alternative Flow |  |
| Exception Flow | 5a. Nếu gửi xác nhận không thành công:  5a1. Hệ thống thông báo lỗi gửi xác nhận.  5a2. Use Case không thực hiện thành công. |

### 2.5.3 Usecase Manage Appointment

|  |  |
| --- | --- |
| Use Case ID | UC-3.1 |
| Use Case Name | Create Repair Appointment |
| Description | Là Admin, tôi muốn tạo mới lịch hẹn sửa chữa cho khách hàng để lưu thông tin lịch hẹn vào hệ thống. |
| Actor(s) | Admin |
| Trigger | Admin chọn “Tạo mới lịch hẹn” từ giao diện. |
| Pre-Condition(s): | * Hệ thống đang hoạt động bình thường. * Admin đã đăng nhập hệ thống. |
| Post-Condition(s): | * Lịch hẹn sửa chữa mới được lưu trữ trong hệ thống. |
| Basic Flow | 1. Admin chọn “Tạo mới lịch hẹn”. 2. Hệ thống hiển thị form nhập thông tin. 3. Admin điền thông tin lịch hẹn và gửi. 4. Hệ thống kiểm tra và lưu vào DB. 5. Hệ thống xác nhận lưu thành công và thông báo cho Admin. |
| Alternative Flow | 3a. Nếu thông tin không hợp lệ, hệ thống yêu cầu nhập lại. |
| Exception Flow | 5a. Nếu lưu thất bại, hệ thống hiển thị lỗi. |

|  |  |
| --- | --- |
| Use Case ID | UC-3.2 |
| Use Case Name | Search Repair Appointment |
| Description | Là Admin, tôi muốn tìm kiếm lịch hẹn sửa chữa trong hệ thống dựa trên các tiêu chí tìm kiếm như tên khách hàng, biển số xe, ngày hẹn, để nhanh chóng tra cứu thông tin lịch hẹn. |
| Actor(s) | Admin |
| Trigger | Admin chọn tính năng “Tìm kiếm lịch hẹn” từ giao diện quản lý. |
| Pre-Condition(s): | * Hệ thống đang hoạt động bình thường. * Admin đã đăng nhập hệ thống. |
| Post-Condition(s): | * Hệ thống hiển thị danh sách kết quả lịch hẹn tìm kiếm phù hợp. |
| Basic Flow | 1. Admin chọn “Tìm kiếm lịch hẹn”. 2. Hệ thống hiển thị form nhập tiêu chí tìm kiếm (tên khách hàng, biển số xe, ngày hẹn...). 3. Admin nhập thông tin tìm kiếm và gửi. 4. Hệ thống xác nhận tiêu chí tìm kiếm hợp lệ. 5. Hệ thống truy vấn dữ liệu lịch hẹn trong DB theo tiêu chí đã nhập. 6. Hệ thống hiển thị danh sách kết quả phù hợp cho Admin. 7. Admin chọn lịch hẹn từ danh sách kết quả để xem chi tiết hoặc thực hiện các thao tác khác. |
| Alternative Flow | 3a. Nếu tiêu chí tìm kiếm không hợp lệ:  3a1. Hệ thống yêu cầu Admin nhập lại thông tin hợp lệ.  3a2. Admin nhập lại và tiếp tục từ bước 3.  6a. Nếu không có kết quả phù hợp:  6a1. Hệ thống thông báo: “Không tìm thấy lịch hẹn phù hợp.” |
| Exception Flow | 5a. Nếu lưu thất bại, hệ thống hiển thị lỗi. |

|  |  |
| --- | --- |
| Use Case ID | UC-3.3 |
| Use Case Name | Update Repair Appointment |
| Description | Là Admin, tôi muốn chỉnh sửa thông tin lịch hẹn sửa chữa để đảm bảo dữ liệu luôn chính xác và phản ánh đúng nhu cầu của khách hàng. |
| Actor(s) | Admin |
| Trigger | Admin chọn chức năng “Cập nhật lịch hẹn” từ giao diện quản lý. |
| Pre-Condition(s): | * Lịch hẹn cần cập nhật tồn tại trong hệ thống. * Admin đã đăng nhập hệ thống. |
| Post-Condition(s): | * Lịch hẹn được cập nhật thành công trong hệ thống. * Hệ thống phản hồi trạng thái và hiển thị thông báo cho Admin. |
| Basic Flow | 1. Admin chọn “Cập nhật lịch hẹn”. 2. Hệ thống hiển thị form với thông tin lịch hẹn đã có sẵn. 3. Admin chỉnh sửa thông tin cần thay đổi và gửi thông tin cập nhật. 4. Hệ thống kiểm tra dữ liệu. 5. Nếu dữ liệu hợp lệ, hệ thống lưu thông tin mới vào DB. 6. Hệ thống hiển thị thông báo cập nhật thành công. |
| Alternative Flow | 4a. Nếu dữ liệu không hợp lệ:  4a1. Hệ thống thông báo lỗi nhập dữ liệu.  4a2. Admin nhập lại dữ liệu và gửi.  6a. Nếu lưu dữ liệu thất bại:  6a1. Hệ thống thông báo lỗi lưu dữ liệu. |
| Exception Flow | 5a. Nếu xảy ra lỗi hệ thống  5a1. Hệ thống thông báo lỗi và yêu cầu thử lại sau.  5a2. Use Case kết thúc. |

|  |  |
| --- | --- |
| Use Case ID | UC-3.4 |
| Use Case Name | Delete Repair Appointment |
| Description | Là Admin, tôi muốn xóa lịch hẹn sửa chữa khỏi hệ thống để loại bỏ các lịch hẹn không còn giá trị hoặc đã bị hủy. |
| Actor(s) | Admin |
| Trigger | Admin chọn chức năng “Xóa lịch hẹn” từ giao diện quản lý. |
| Pre-Condition(s): | * Lịch hẹn cần xóa tồn tại trong hệ thống. * Admin đã đăng nhập hệ thống. |
| Post-Condition(s): | * Lịch hẹn được thay đổi trạng thái thành “Đã hủy” trong hệ thống. * Hệ thống phản hồi trạng thái và hiển thị thông báo cho Admin. |
| Basic Flow | 1. Admin chọn “Xóa lịch hẹn”. 2. Hệ thống hiển thị xác nhận xóa. 3. Admin xác nhận xóa. 4. Hệ thống kiểm tra điều kiện (trạng thái lịch hẹn). 5. Nếu điều kiện hợp lệ, hệ thống cập nhật trạng thái lịch hẹn thành “Đã hủy” trong DB. 6. Hệ thống hiển thị thông báo xóa thành công. |
| Alternative Flow | 3a. Nếu Admin không xác nhận xóa:  3a1. Hệ thống hủy thao tác xóa.  3a2. Quay lại giao diện quản lý lịch hẹn.  4a. Nếu lịch hẹn ở trạng thái khóa không thể xóa:  4a1. Hệ thống thông báo lỗi: “Lịch hẹn đang bị khóa, không thể xóa.”  4a2. Quay lại giao diện quản lý lịch hẹn. |
| Exception Flow | 5a. Nếu xảy ra lỗi hệ thống  5a1. Hệ thống thông báo lỗi và yêu cầu thử lại sau.  5a2. Use Case kết thúc. |

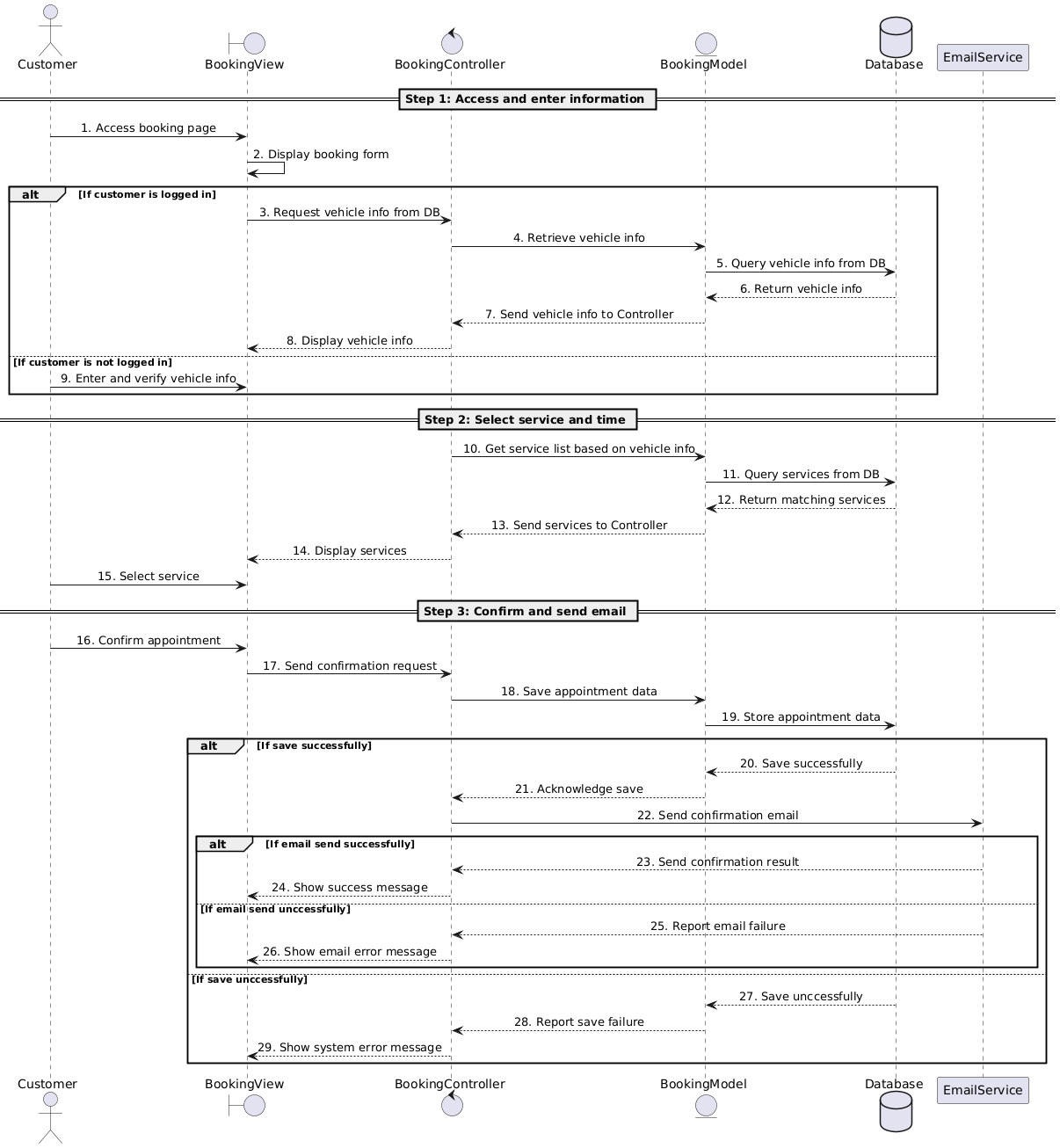
## 2.6 Biểu đồ quy trình đặt lịch

## 

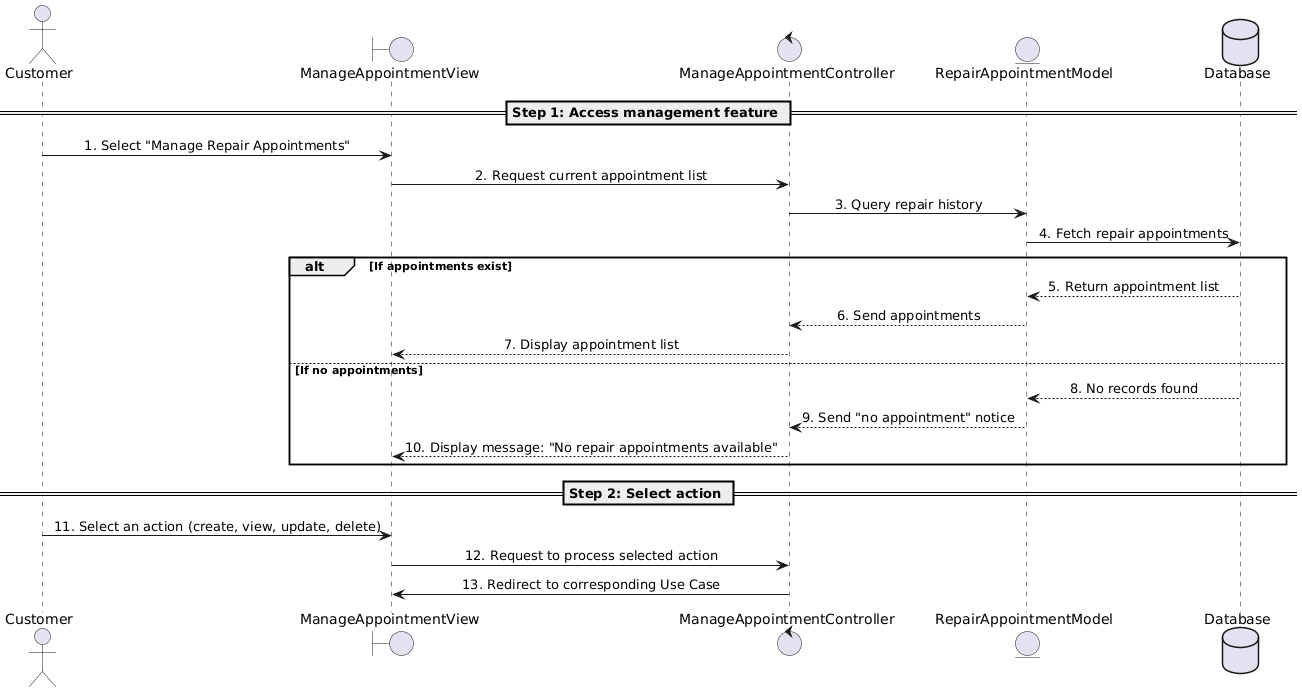
Hình 2: Quy trình nghiệp vụ

## Chương 3: Biểu đồ phân tích hệ thống

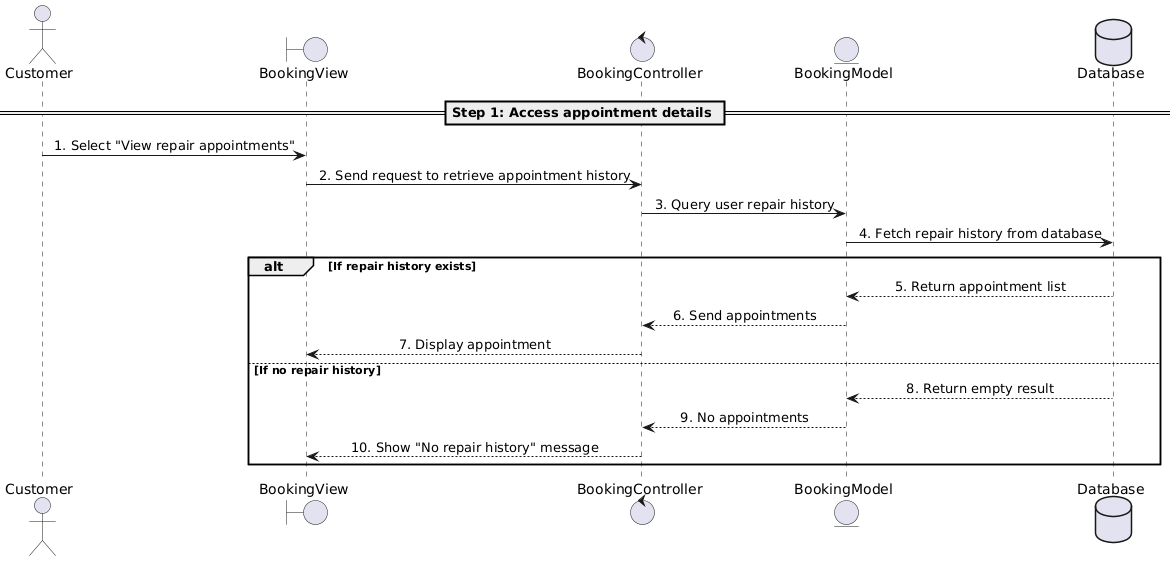
## 3.1. Biểu đồ tuần tự



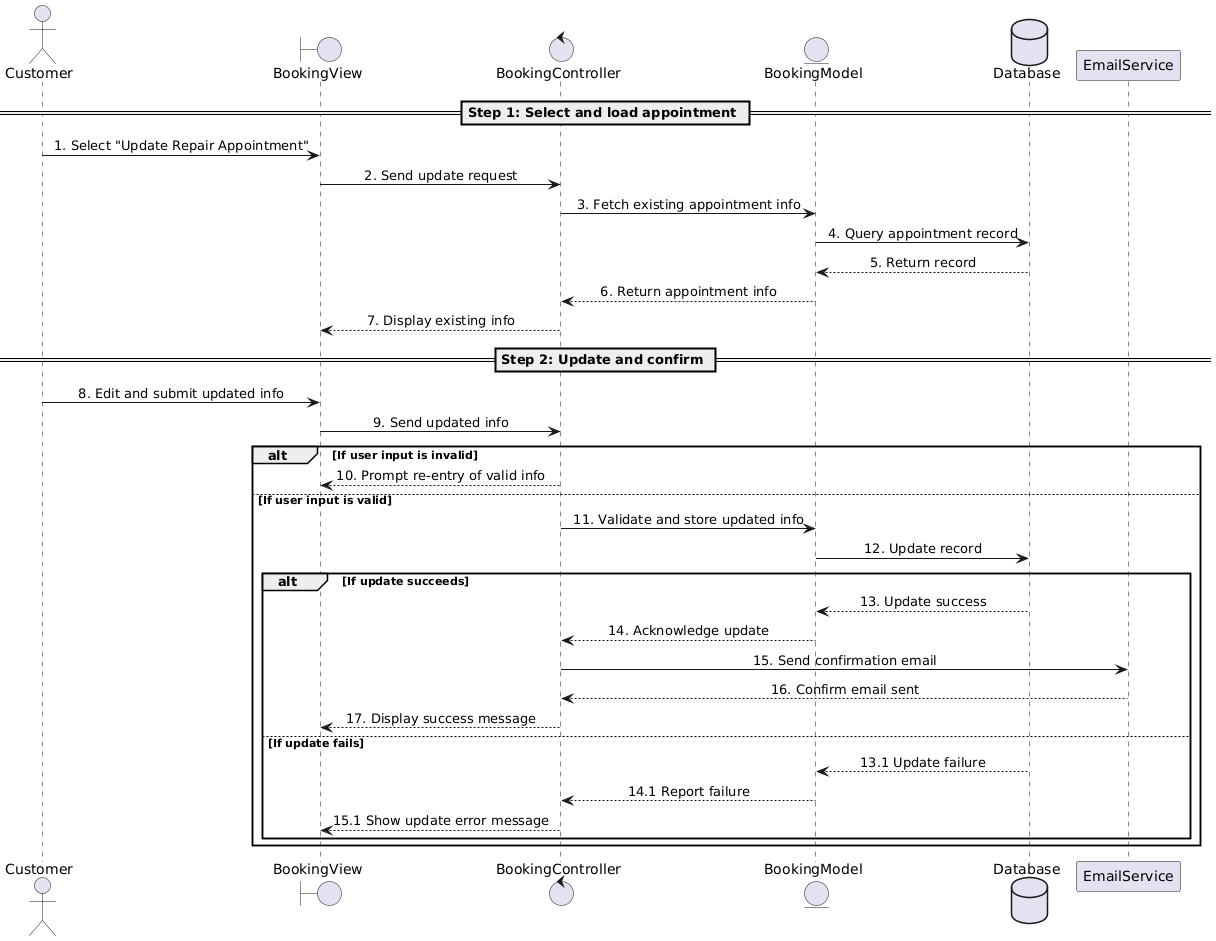
Hình 3: Biểu đồ trình tự Booking Repair Appointment



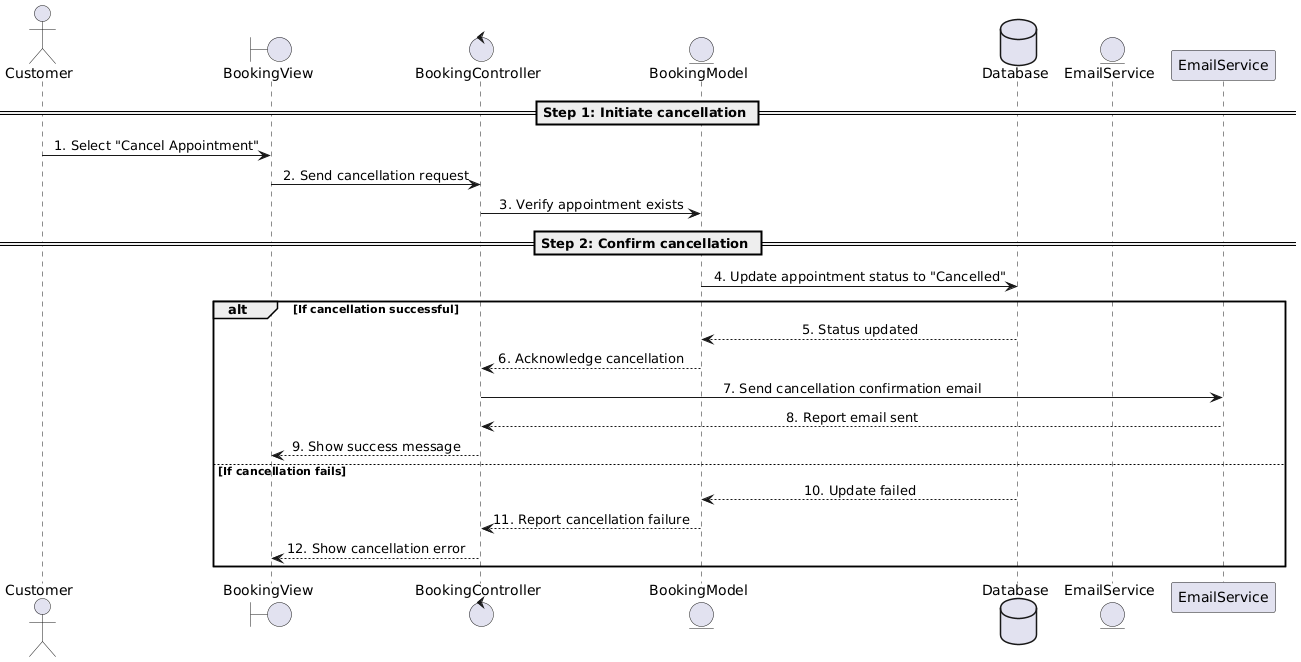
Hình 4: Biểu đồ trình tự Manage Repare Appointment



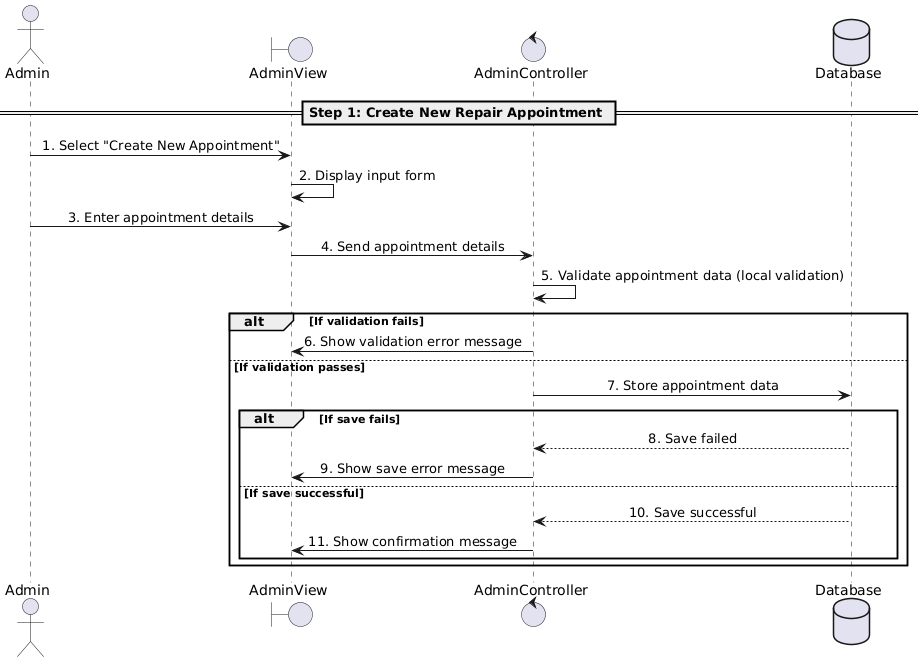
Hình 5: Biểu đồ trình tự View Repair Appointment



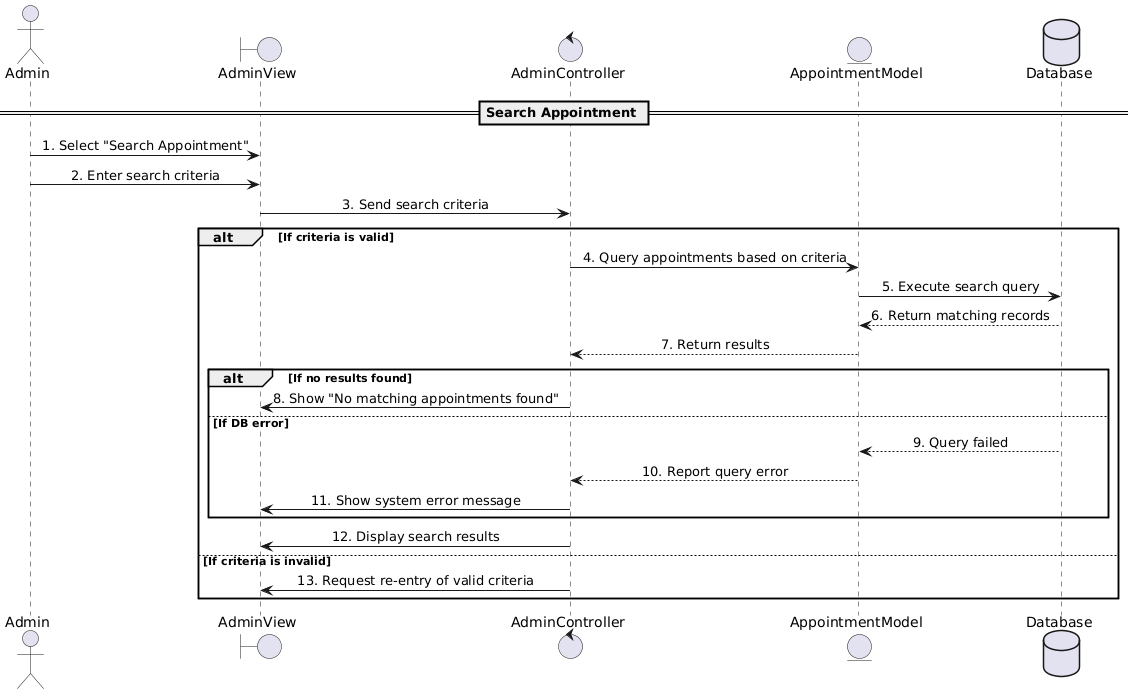
Hình 6: Biểu đồ trình tự Update Repair Appointment



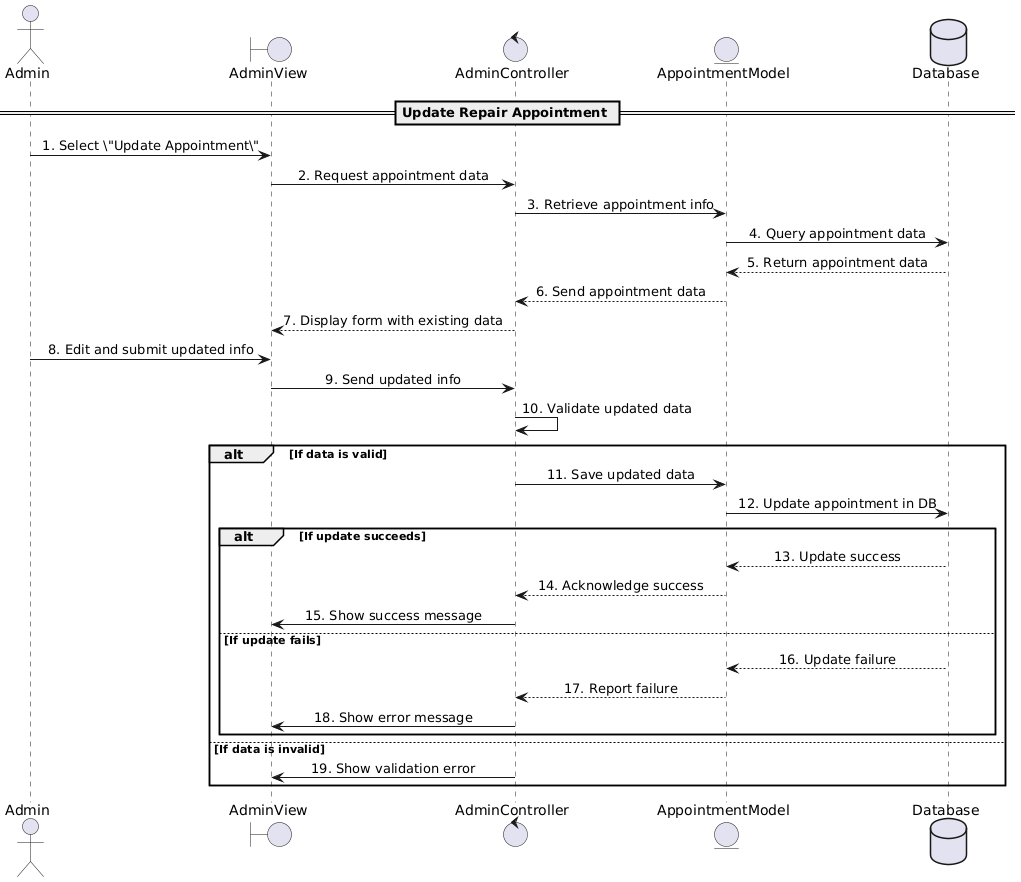
Hình 7: Biểu đồ trình tự Cancel Repair Appointment



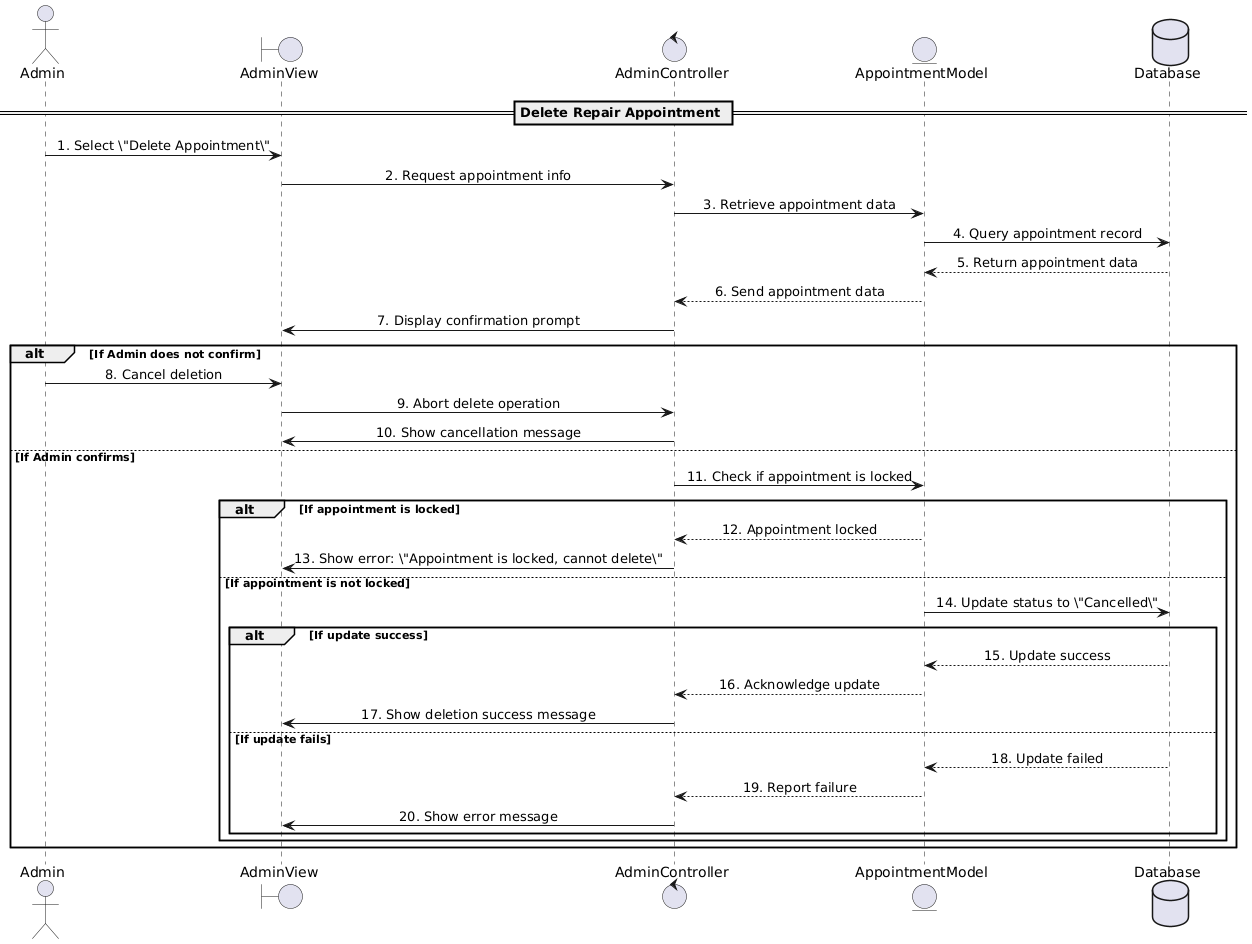
Hình 8: Biểu đồ trình tự Create Repair Appointment ( Admin)



Hình 9: Biểu đồ trình tự Search Repair Appointment (Admin)

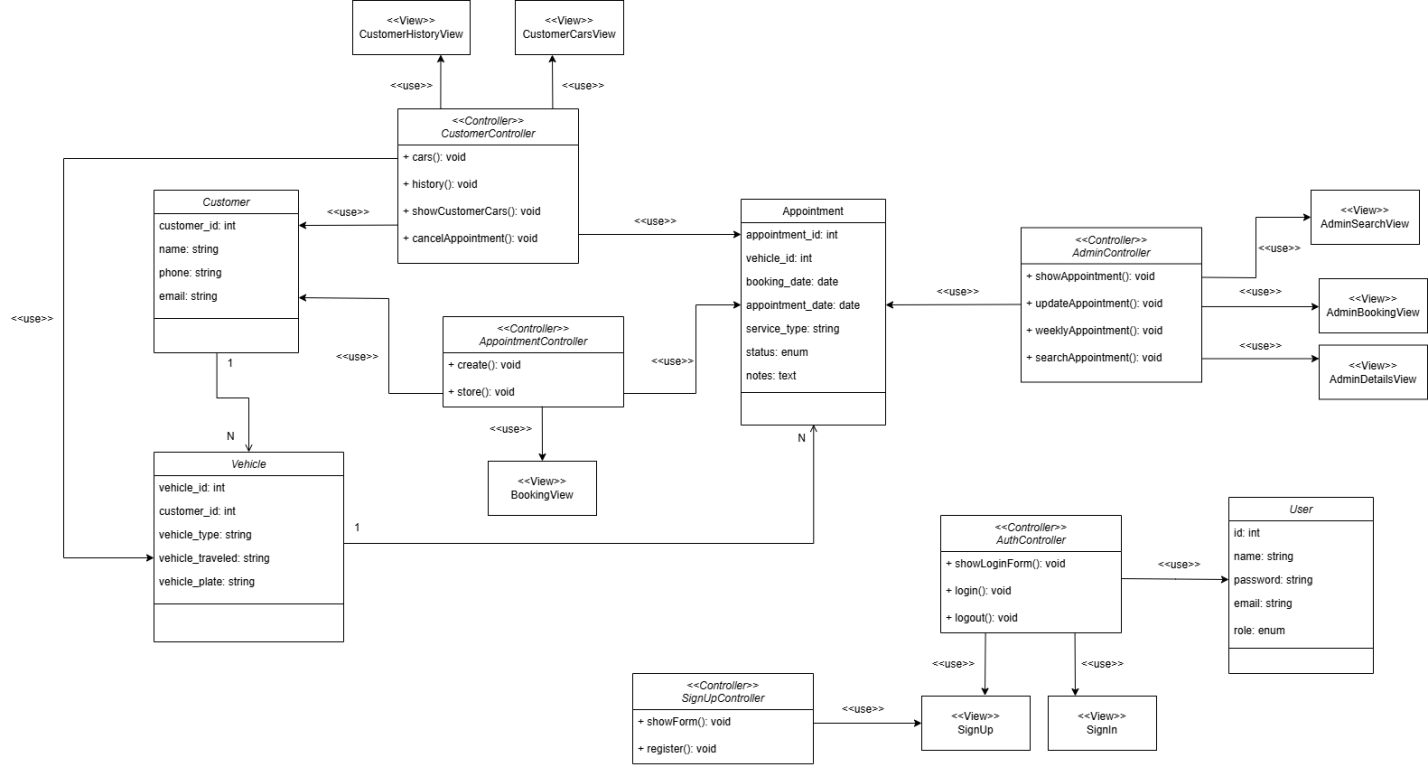


Hình 10: Biểu đồ trình tự Update Repair Appointment ( Admin)



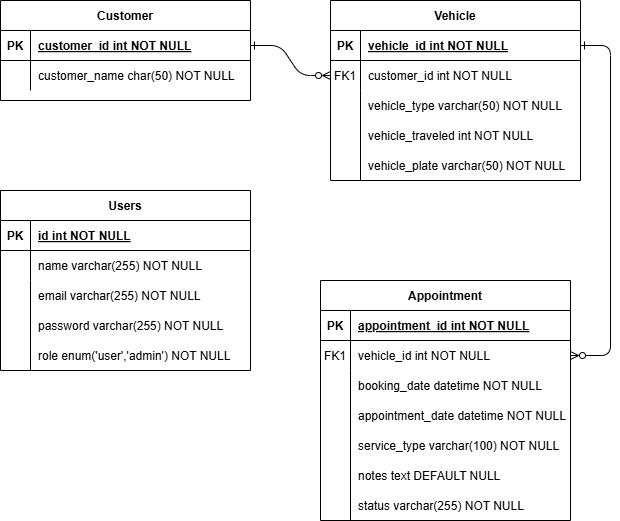
Hình 11: Biểu đồ trình tự Delete Repair Appointment (Admin)

## 3.2. Thiết kế lớp (Class Diagram)



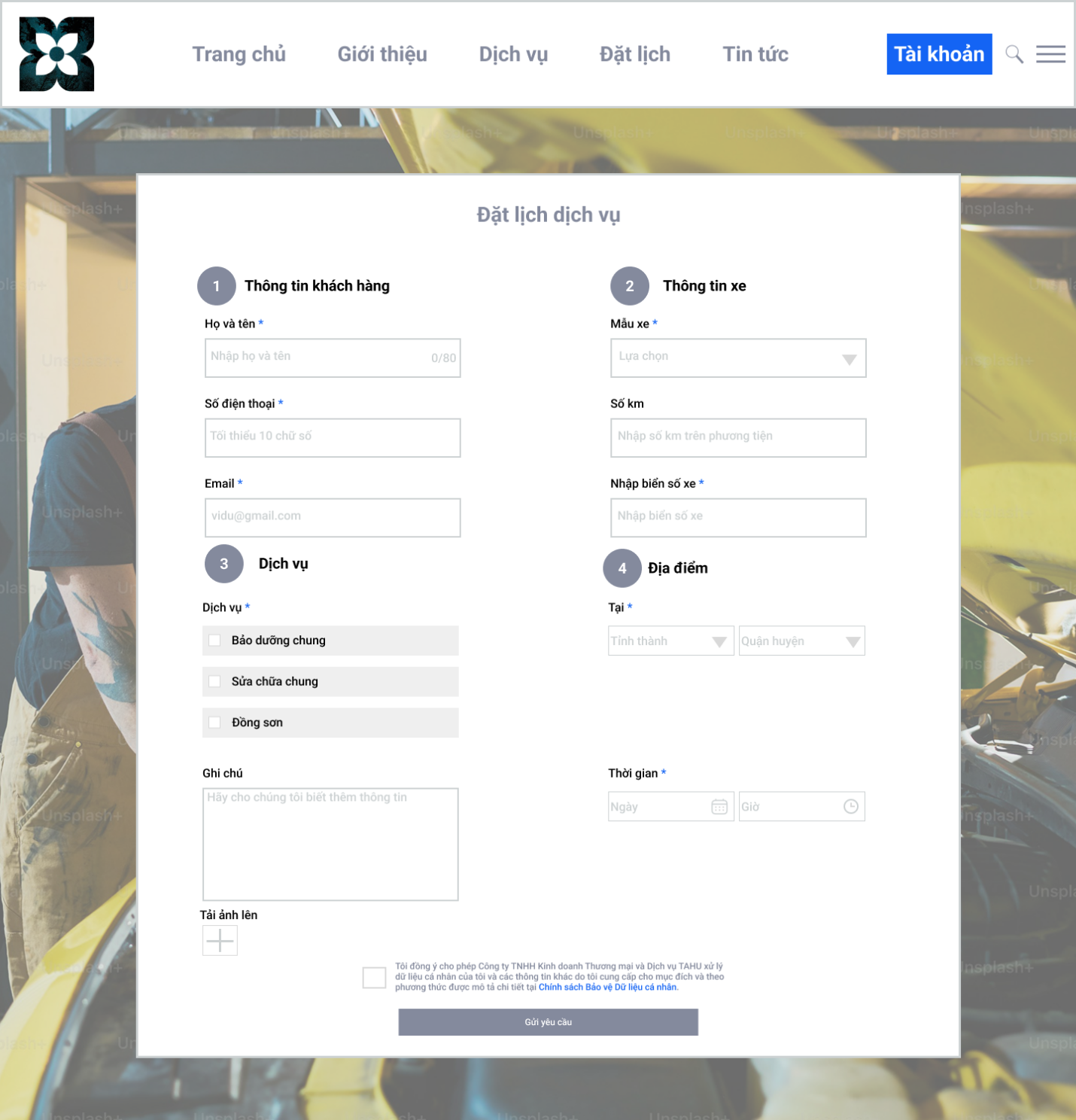
Hình 12: Biểu đồ lớp

## 3.3. Thiết kế cơ sở dữ liệu

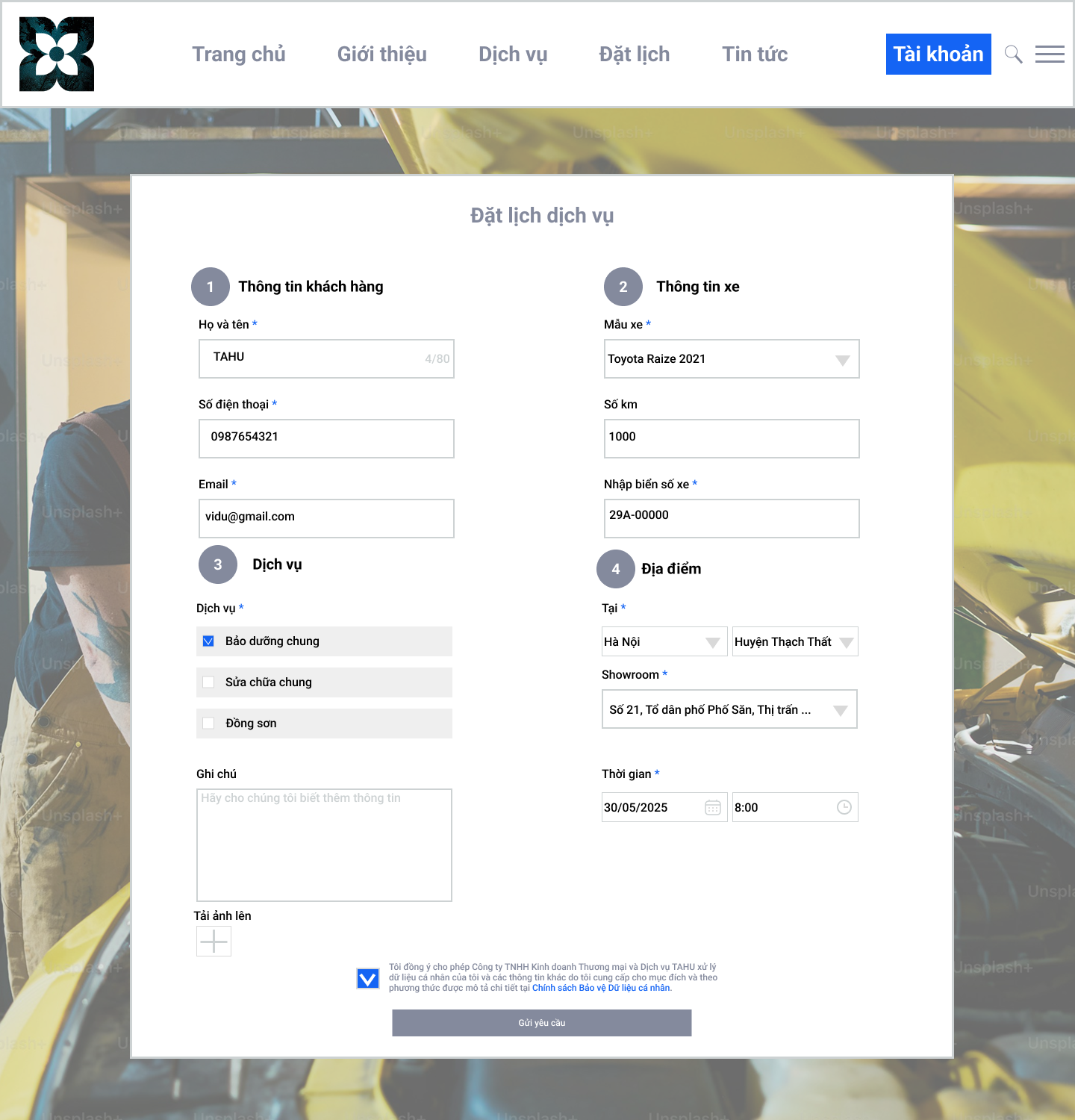


Hình 13: Biểu đồ ERD

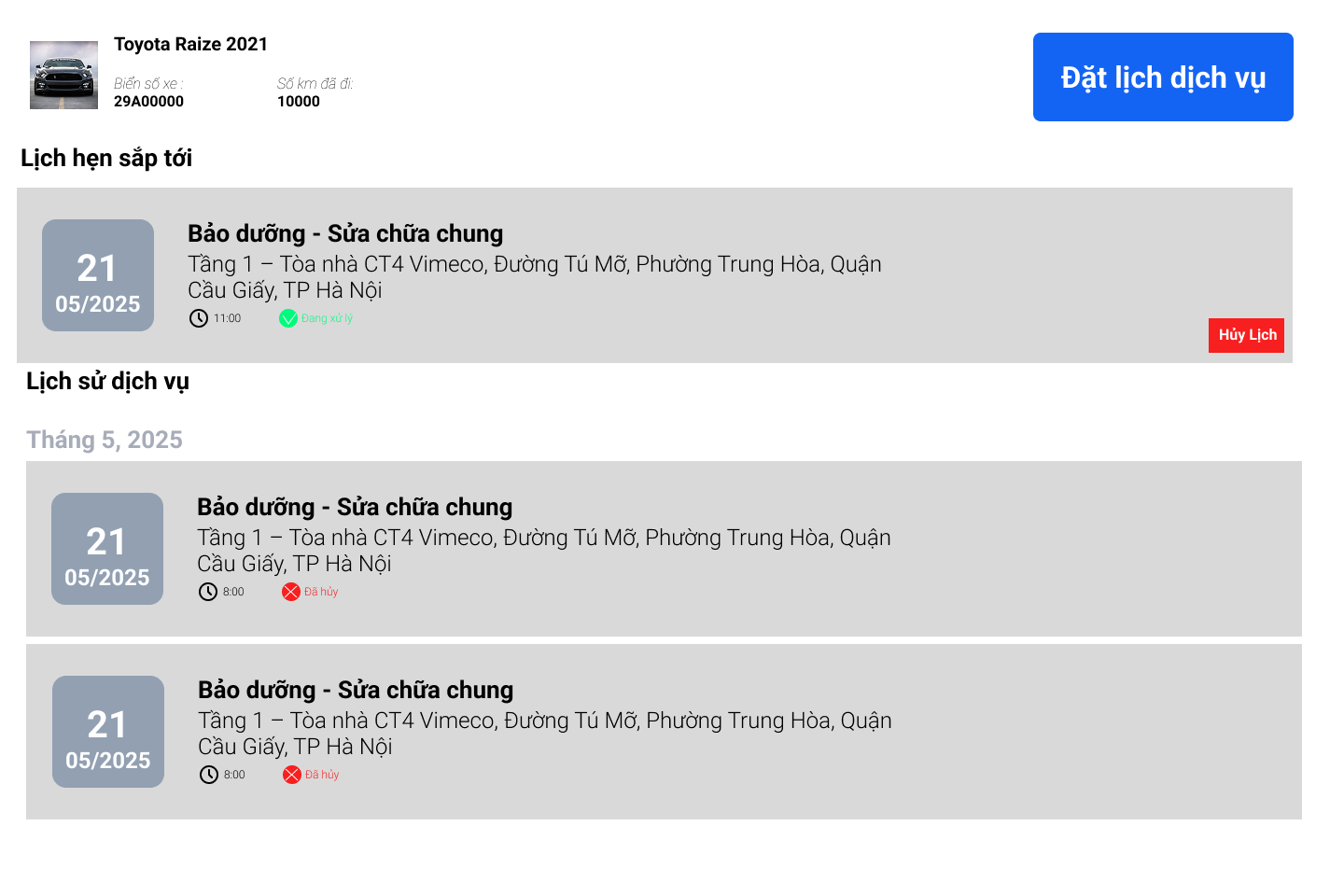
## 3.4. Thiết kế giao diện người dùng



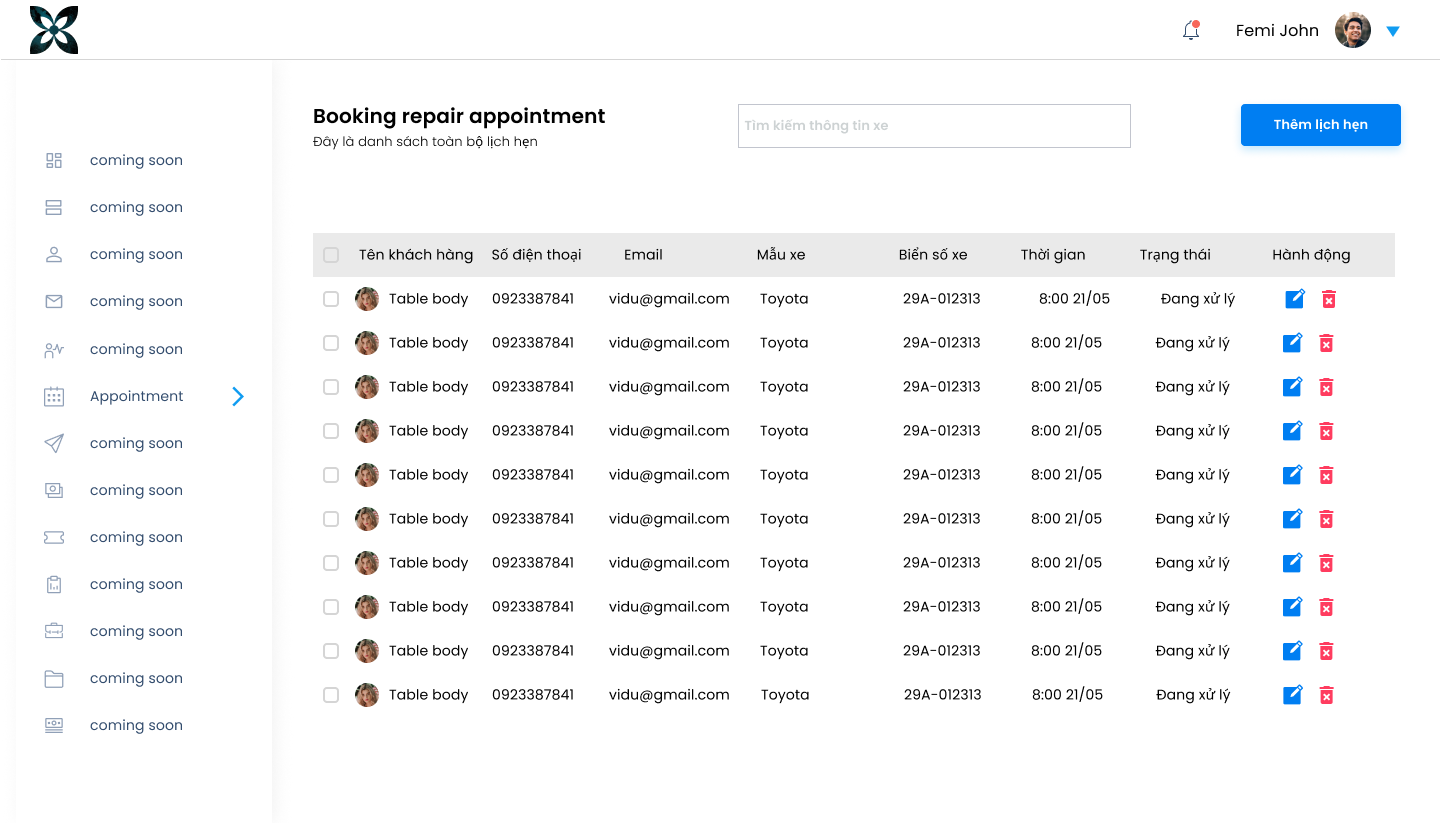
Hình 14 Giao diện đặt lịch



Hình 15: Giao diện người dùng nhập thông tin đặt lịch



Hình 16: Giao diện theo dõi lịch hẹn phía người dùng



Hình 17: Giao diện admin

# Chương 4. Cài đặt và triển khai hệ thống

## 4.1. Môi trường triển khai

Hệ thống được triển khai trên môi trường sau:

**Phần cứng:**

Máy chủ chạy backend: CPU Intel Core i5 trở lên, RAM 8GB, ổ cứng SSD 256GB trở lên.

Máy tính hoặc thiết bị người dùng truy cập hệ thống qua trình duyệt web.

**Phần mềm:**

Hệ điều hành: Ubuntu 20.04 LTS hoặc Windows 10

Web server: Xampp.

Backend: PHP 8.x, Laravel Framework 9.x.

Frontend: Node.js 16.x, React 20.x.

Cơ sở dữ liệu: MySQL 8.x hoặc dùng PhPMyAdmin

Trình duyệt: Chrome, Firefox, Edge (phiên bản mới nhất).

**Mạng:**

Kết nối Internet ổn định, băng thông tối thiểu 10 Mbps cho máy chủ.

## 4.2. Cài đặt hệ thống

### 4.2.1 Yêu cầu hệ thống

Máy chủ cần được cài đặt các phần mềm cần thiết như PHP, Composer, MySQL, Node.js, npm.

Môi trường phát triển frontend (React) và backend (Laravel) đã được thiết lập.

Tài khoản quản trị cơ sở dữ liệu có quyền tạo và quản lý database.

Trình duyệt web được hỗ trợ trên máy người dùng để truy cập giao diện frontend.

### 4.2.2 Các bước cài đặt chương trình

**Bước 1: Cài đặt backend Laravel**

Cài đặt PHP, Composer trên máy chủ.

Clone hoặc tải source code backend từ kho lưu trữ.

Chạy lệnh composer install để cài đặt các thư viện phụ thuộc.

Sao chép file .env.example thành .env và cấu hình kết nối database, các biến môi trường cần thiết.

Chạy lệnh php artisan key:generate để tạo khóa ứng dụng.

Thực hiện migrate database bằng lệnh php artisan migrate.

Khởi động server Laravel bằng lệnh php artisan serve và cấu hình Xampp phục vụ ứng dụng.

**Bước 2: Kết nối và kiểm thử**

Đảm bảo backend và frontend kết nối với nhau qua API routes.

Kiểm tra các chức năng cơ bản như đăng ký, đăng nhập, tìm kiếm sân, đặt sân và thanh toán.

Kiểm tra các quyền truy cập của người dùng và chủ sân.

Kiểm thử tính ổn định và hiệu năng hệ thống.

# Chương 5. Kết quả và đánh giá

## 5.1. Kết quả thử nghiệm hệ thống

Sau khi hoàn thiện việc phát triển và triển khai, hệ thống Quản lý Garage và Đặt Lịch Hẹn Online đã được tiến hành thử nghiệm với các kịch bản sau:

* **Thử nghiệm đặt lịch hẹn:** Người dùng dễ dàng đặt lịch sữa chữa theo đúng nhu cầu, loại xe và dịch vụ; đặt lịch hẹn theo ngày giờ mong muốn và nhận được xác nhận nhanh chóng thông qua email.
* **Thử nghiệm quản lý lịch đặt:** Người dùng có thể xem, hủy và thay đổi lịch hẹn đúng quy định và nhanh chóng
* **Thử nghiệm quản lý garage:** Chủ garage dễ dàng thêm, chỉnh sửa lịch hẹn, xem và tìm kiếm tình trạng lịch hẹn.
* **Kiểm thử hiệu năng:** Các thao tác trên hệ thống được xử lý ổn, thời gian phản hồi dưới 5 giây trong môi trường thử nghiệm với số lượng người dùng vừa phải.

## 5.2. Đánh giá hiệu quả hệ thống

* **Ưu điểm:**
  + Hệ thống giúp số hóa quy trình quản lý lịch hẹn và đặt lịch, giảm thiểu sai sót so với phương pháp thủ công.
  + Tối ưu trải nghiệm người dùng với giao diện trực quan, thao tác đặt lịch hẹn nhanh chóng và nhận được thông báo xác nhận qua Email.
  + Hỗ trợ chủ Garage theo dõi và quản lý lịch đặt chính xác, giúp nâng cao hiệu quả kinh doanh.
  + Đảm bảo tính minh bạch qua chức năng đánh giá và phản hồi, tăng độ tin cậy cho khách hàng.
* **Hạn chế:**
  + Hệ thống hiện chưa hỗ trợ ứng dụng trên di động (Mobile App), có thể hạn chế đối tượng người dùng.
  + Chưa áp dụng các công nghệ tối ưu hóa hiệu năng cho lượng người dùng rất lớn.
  + API khoảng cách còn phụ thuộc vào kết nối mạng nhanh/chậm rất nhiều.
* **Đề xuất cải tiến:**
  + Phát triển ứng dụng mobile để tăng khả năng tiếp cận khách hàng.
  + Mở rộng các các chức năng như: Quản lý khách hàng, thống kê doanh thu và tích hợp các dịch vụ hỗ trợ khác.
  + Nâng cao khả năng xử lý hiệu năng và bảo mật cho quy mô lớn hơn.
  + Bổ sung tính năng tương tác nâng cao như chatbot hỗ trợ khách hàng.