BÌA

**MỤC** **LỤC**

[DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT 1](#_Toc167027595)

[DANH MỤC HÌNH ẢNH 3](#_Toc167027596)

[LỜI NÓI ĐẦU 7](#_Toc167027597)

[CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI 8](#_Toc167027598)

[**2.1. Hệ quản trị cơ sơ dữ liệu MySQL** 10](#_Toc167027599)

[2.1.1. Khái niệm 10](#_Toc167027600)

[2.1.2. Đặc điểm 10](#_Toc167027601)

[**2.2. Tổng quan về Java** 14](#_Toc167027602)

[2.2.1. Khái niệm 14](#_Toc167027603)

[2.2.2.. Đặc điểm của Java 15](#_Toc167027604)

[2.2.3. Các platform của Java 16](#_Toc167027605)

[2.2.4. Tiêu chuẩn của một môi trường Java điển hình 16](#_Toc167027606)

[2.3. Tổng quan về ReactJS 17](#_Toc167027607)

[**2.4. Tổng quan về StarUML** 19](#_Toc167027608)

[CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 22](#_Toc167027609)

[**3.1. Các bước hoàn thành dự án** 22](#_Toc167027610)

[3.2.1. Tác nhân và ca sử dụng 23](#_Toc167027611)

[3.2.2. Use case 23](#_Toc167027612)

[3.2.3. Đặc tả ca sử dụng 24](#_Toc167027613)

[**Biểu đồ ca sử dụng tham gia quản lý bãi đỗ xe** 26](#_Toc167027614)

[**Biểu đồ trình tự** 28](#_Toc167027615)

[3.4.1. Quản lý bãi đỗ 28](#_Toc167027616)

[3.4.2. Quản lý khách hàng 29](#_Toc167027617)

[**3.2.4.** Quản lý thẻ tháng 30](#_Toc167027618)

[**3.2.5.** Quản lý tài khoản 31](#_Toc167027619)

[**3.3.** **Biểu đồ trạng thái** 32](#_Toc167027620)

[3.5.1. Tài khoản 32](#_Toc167027621)

[3.5.2. Vị trí đặt (Parking slot) 32](#_Toc167027622)

[3.5.3. Thẻ tháng (Regular pass) 33](#_Toc167027623)

[**Biểu đồ hoạt động** 34](#_Toc167027624)

[3.6.1. Chức năng đăng nhập 34](#_Toc167027625)

[3.5.4. Chức năng đăng kí 35](#_Toc167027626)

[3.5.5. Chức năng tạo bãi đỗ xe 36](#_Toc167027627)

[3.5.6. Chức năng đặt chỗ 37](#_Toc167027628)

[3.5.7. Chức năng đăng kí thẻ tháng 38](#_Toc167027629)

[3.5.8. Chức năng cập nhật thông tin cá nhân 39](#_Toc167027630)

[**Biểu đồ lớp chi tiết** 40](#_Toc167027631)

[3.6. **Biểu đồ thành phần** 41](#_Toc167027632)

[3.8.1. Khách hàng 41](#_Toc167027633)

[3.6.2. Admin 41](#_Toc167027634)

[3.7. **Thiết kế cơ sở dữ liệu** 43](#_Toc167027635)

[3.9.1. Cơ sở dữ liệu và mối quan hệ 43](#_Toc167027636)

[3.7.2. Mô tả chi tiết các bảng 43](#_Toc167027637)

[CHƯƠNG 4: KẾT QUẢ CÀI ĐẶT CHƯƠNG TRÌNH 48](#_Toc167027638)

[4.1. Chức năng dành cho người thuê bãi đỗ xe 48](#_Toc167027639)

[4.2. Chức năng dành cho admin 53](#_Toc167027640)

[4.2.1. Giao diện chính 53](#_Toc167027641)

[4.2.1. Giao diện quản lý khách hàng 54](#_Toc167027642)

[4.2.2. Giao diện thêm khách hàng 55](#_Toc167027643)

[4.2.2. Giao diện quản lý bãi đỗ 56](#_Toc167027644)

[4.2.2. Giao diện quản lý bãi đỗ 57](#_Toc167027645)

# DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **Thuật ngữ/ Từ viết tắt** | **Cụm từ đầy đủ** | **Ý nghĩa** |
| API | Application Programming Interface | Giao diện lập trình ứng dụng |
| CSDL | Cơ sở dữ liệu | Cơ sở dữ liệu |
| AES | Advanced Encryption Standard | Tiêu chuẩn mã hoá tiên tiến |
| BMP | Bitmap | Ảnh không nén Bitmap |
| DOM | Document Object Model | Giao diện lập trình ứng dụng |
| CPU | Central Processing Unit | Bộ xử lý trung tâm |
| CI | Continuous Integration | Tích hợp liên tục |
| DOM | Document Object Model | Mô hình các đối tượng trong tài liệu HTML |
| EDR | Endpoint Detection and Response | Phát hiện và phản hồi điểm cuối |
| Framework | Framework | Framework là phần mềm được phát triển và sử dụng bởi các nhà phát triển để xây dựng  các ứng dụng |
| GUI | Graphical User Interface | Giao diện đồ họa người dùng |
| HTML | HyperText Markup  Language | Ngôn ngữ đánh dấu siêu văn bản |
| IDE | Integrated Development  Environment | Môi trường phát triển tích hợp |
| IE | Internet Explorer | Một trình duyệt web |
| Module | Module | Một thành phần phần mềm hoặc một phần của chương trình |
| TCP | Transmission Control Protocol | Giao thức điều khiển truyền tin |
| UDP | User Datagram Protocol | Giao thức giao vận không kết nối |
| SSL | Secure Socket Layer | Giao thức cổng an toàn |

# DANH MỤC HÌNH ẢNH

[2.1.3. Lợi ích khi sự dụng MySQL 11](#_Toc167026664)

[Hình 2.1: Lợi ích khi sử dụng MySQL 11](#_Toc167026665)

[Hình 2.2: Platform cơ bản của Java 16](#_Toc167026666)

[Hình 2.3: Môi trường java điển hình 17](#_Toc167026667)

[Hình 3.1. Biểu đồ ca sử dụng của admin 26](#_Toc167026668)

[Hình 3.6: Biểu đồ trình tự bãi đỗ 29](#_Toc167026669)

[Hình 3.7: Biểu đồ trình tự khách hàng 29](#_Toc167026670)

[Hình 3.8: Biểu đồ trình tự thẻ tháng 30](#_Toc167026671)

[Hình 3.11: Biểu đồ trình tự quản lý tài khoản 31](#_Toc167026672)

[Hình 3.13. Biểu đồ trạng thái của tài khoản 32](#_Toc167026673)

[Hình 3.14: Biểu đồ trạng thái vị trí đặt 33](#_Toc167026674)

[Hình 3.15: Biểu đồ trạng thái của thẻ tháng 34](#_Toc167026675)

[Hình 3.17: Biểu đồ hoạt động của đăng nhập 35](#_Toc167026676)

[Hình 3.18: Biểu đồ hoạt động của đăng kí 35](#_Toc167026677)

[Hình 3.19: Biểu đồ hoạt động tạo bãi đỗ xe 36](#_Toc167026678)

[Hình 3.20: Biểu đồ hoạt động của chức năng đặt chỗ 38](#_Toc167026679)

[Hình 3.21: Biểu đồ hoạt động của đăng kí thẻ tháng 39](#_Toc167026680)

[Hình 3.22: Biểu đồ hoạt động của chức năng cập nhật thông tin 40](#_Toc167026681)

[Hình 3.28: Biều đồ lớp chi tiết 41](#_Toc167026682)

[Hình 3.29: Biểu đồ thành phần người quản trị 41](#_Toc167026683)

[Hình 3.30: Biểu đồ thành phần admin 42](#_Toc167026684)

[Hình 3.31: Cơ sở dữ liệu và mối quan hệ 43](#_Toc167026685)

[Hình 3.32: Hình ảnh giao diện chính 49](#_Toc167026686)

[Hình 3.33: Hình ảnh giao diện đặt vé tháng 50](#_Toc167026687)

[Hình 3.34: Hình ảnh giao diện đặt chỗ đỗ xe 51](#_Toc167026688)

[Hình 3.35: Hình ảnh giao diện cập nhật thông tin tài khoản 51](#_Toc167026689)

[Hình 3.36: Hình ảnh giao diện đổi mật khẩu 52](#_Toc167026690)

[Hình 3.37: Hình ảnh giao diện chính của admin 53](#_Toc167026691)

[Hình 3.38: Hình ảnh giao diện quản lý khách hàng 55](#_Toc167026692)

[Hình 3.39: Hình ảnh giao diện thêm khách hàng 56](#_Toc167026693)

[3.41: Hình ảnh giao diện các thao tác chính của bãi đỗ xe 57](#_Toc167026694)

**LỜI CẢM ƠN**

Để hoàn thành đồ án, trước hết em xin gửi lời cảm ơn chân thành và sâu sắc đến cô giáo ….., người đã tận tình hướng dẫn em trong suốt quá trình nghiên cứu vừa qua. Trong thời gian được cô hướng dẫn, em không những tiếp thu thêm nhiều kiến thức bổ ích mà còn học tập được tinh thần, thái độ làm việc nghiêm túc, hiệu quả. Đây là điều rất cần thiết cho em trong quá trình học tập và làm việc sau này.

Em chân thành cảm ơn các thầy cô trong khoa công nghệ thông tin, …… đã tâm huyết dạy dỗ, truyền đạt những kiến thức quý báu cho em trong suốt 5 năm học đại học. Những kiến thức đó không chỉ giúp em hoàn thành đồ án tốt nghiệp này mà còn là thứ hành trang quý báu để em có thể tự tin trên quá trình theo đuổi thành công của mình.

Xin kính chúc tất cả mọi người dồi dào sức khỏe và tràn đầy nhiệt huyết trong cuộc sống cũng như công việc của mình.

**Sinh viên thực hiện đồ án**

# LỜI NÓI ĐẦU

Khi xã hội ngày càng phát triển và dân số đông thì nhu cầu nhà ở ngày càng một tăng cao. Đặc biệt với nhóm đối tượng như sinh viên mới nhập học, sinh viên vừa ra trường, vợ chồng vừa kết hôn, người mới đi làm, công nhân,...thì nhu cầu nhà ở của họ là hết sức cần thiết. Nhưng kinh tế là một trở ngại lớn khi hầu như họ không đủ khả năng để mua một căn hộ bởi giá rất đắt đỏ. Thấy được tiềm năng đó, người ta đã xây dựng các khu nhà ở cho thuê (nhà trọ) với mức giá rẻ, phù hợp hơn để đáp ứng thị trường.

Thế nhưng xây dựng là một chuyện, quản lý lại là một chuyện phải cân nhắc. Khi số lượng phòng nhiều và khách trọ ở đông, thì việc quản lý tất cả thông tin phòng trọ, khách thuê trọ, quản lý thu chi tiền điện nước và các dịch vụ phát sinh, quản lý các trang thiết bị của mỗi phòng và đặc biệt thống kê cần được thực hiện một cách nhanh chóng, linh hoạt và chặt chẽ.

Quản lý bằng sổ sách vẫn khả thi, nhưng hiệu suất không cao nếu ta phải xử lý hàng xấp giấy tờ. Do đó, cần có một hệ thống website quản lý và cho thuê nhà trọ để giải quyết vấn đề trên. Giúp chúng ta lưu trữ thông tin an toàn, lâu dài và tự động hóa tối đa các thao tác so với thực tế như tính toán, thống kê.

**SINH VIÊN THỰC HIỆN ĐỒ ÁN**

# CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

**1.1. Mục tiêu của đề tài**

1. Tối ưu hóa quy trình quản lý và cho thuê bãi gửi xe ô tô: Mục tiêu của đề tài là tạo ra một hệ thống website hiệu quả giúp cải thiện quy trình quản lý, giám sát và cho thuê bãi gửi xe, tối ưu hóa thời gian cho việc tìm kiếm và đặt lịch thuê bãi.

2. Xây dựng giao diện người dùng thân thiện và dễ sử dụng: Đề tài nhằm tạo ra giao diện người dùng tương tác trực quan, dễ dàng sử dụng và đáp ứng nhu cầu của cả người quản lý và người sử dụng dịch vụ thuê.

3. Tăng cường tính bảo mật và quyền riêng tư: Đảm bảo an toàn thông tin và quyền riêng tư của người dùng thông qua việc áp dụng các biện pháp bảo mật hiện đại và chuẩn mực bảo vệ dữ liệu.

4. Phát triển chức năng tối ưu hóa tìm kiếm và lọc thông tin: Nghiên cứu và triển khai các tính năng tìm kiếm thông minh, lọc dữ liệu linh hoạt để người dùng dễ dàng tìm kiếm và thuê bãi đỗ xe phù hợp với thời gian và địa điểm mong muốn.

5. Kiểm thử, đánh giá và cải tiến hiệu suất hệ thống: Thực hiện các bài kiểm tra, đánh giá tính năng và hiệu suất của hệ thống để đảm bảo hoạt động ổn định, giảm thiểu lỗi và nâng cao trải nghiệm người dùng.

**1.2. Các bước hoàn thành đề tài**

1. Phân tích yêu cầu và thiết kế cơ sở dữ liệu:

* Nghiên cứu và thu thập yêu cầu từ người dùng, bao gồm admin và người dung cần gửi xe.
* Thiết kế cơ sở dữ liệu phù hợp để lưu trữ thông tin về bãi đỗ xe, tài khoản người dùng, đặt lịch gửi xe, v.v.

2. Xây dựng backend với Java Spring Boot:

* Xây dựng RESTful API để cung cấp dữ liệu và chức năng cho giao diện người dùng.
* Thực hiện xác thực và quản lý đăng nhập người dùng.
* Xử lý các yêu cầu từ giao diện và thực hiện các chức năng quản lý, cho thuê, tìm kiếm, lọc, v.v.

3. Xây dựng frontend với ReactJS:

* Thiết kế giao diện người dùng trực quan và thân thiện.
* Giao tiếp với API backend để truy xuất và hiển thị dữ liệu.
* Tối ưu hóa trải nghiệm người dùng với các tính năng tìm kiếm và lọc thông minh.

4. Cải tiến tính bảo mật và quyền riêng tư:

* Áp dụng các biện pháp bảo mật như mã hóa dữ liệu, xác thực hai yếu tố, bảo vệ khỏi các cuộc tấn công thông thường.
* Đảm bảo dữ liệu cá nhân được bảo vệ và chỉ được truy cập bởi người dùng có quyền.

5. Kiểm thử và đánh giá hiệu suất:

* Thực hiện các bài kiểm tra tính năng và kiểm tra độ tin cậy của hệ thống.
* Đánh giá hiệu suất hệ thống để đảm bảo tốc độ và đáp ứng của trang web là ổn định và tốt.

6. Triển khai và triển khai:

* Triển khai ứng dụng trên môi trường sản xuất.
* Đảm bảo sự ổn định và hiệu suất tốt khi có lượng người dùng thực tế.

**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

## **2.1. Hệ quản trị cơ sơ dữ liệu MySQL**

### 2.1.1. Khái niệm

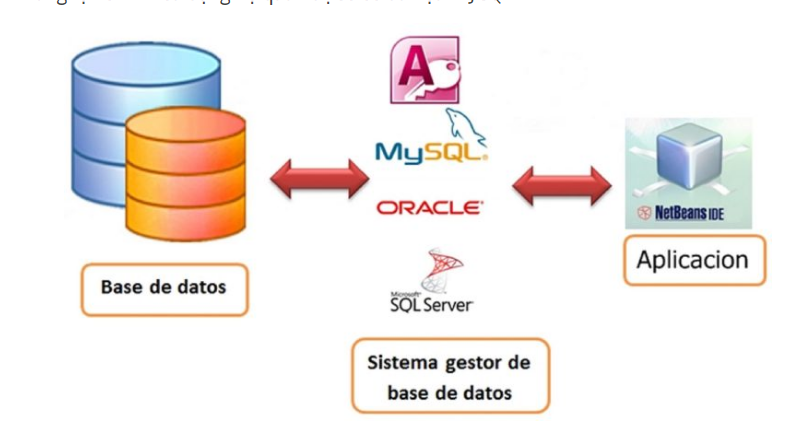
**Hệ quản trị cơ sở dữ liệu MySQL**được hiểu như là chương trình dùng để quản lý hệ thống cơ sở dữ liệu, trong đó, cơ sở dữ liệu là một hệ thống lưu trữ thông tin được sắp xếp rõ ràng, phân lớp ngăn nắp. Nó giúp bạn có thể truy cập dữ liệu một cách thuận lợi và nhanh chóng nhất. Vì hỗ trợ đa số các ngôn ngữ lập trình nên MySQL chính là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất trên thế giới. Hiện MySQL đang được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng.

MySQL là cơ sở dữ liệu tốc độ cao, ổn định hoạt động trên nhiều hệ điều hành, cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Đặc biệt, **hệ quản trị cơ sở dữ liệu MySQL**hoàn toàn miễn phí nên người dùng có thể thoải mái tải về từ trang chủ. Nó có rất nhiều những phiên bản cho các hệ điều hành khác nhau. MySQL được sử dụng cho việc bỗ trợ PHP, [Perl](https://www.perl.org/) và nhiều ngôn ngữ khác. Là nơi lưu trữ những thông tin trên các trang web viết bằng [framework PHP](http://dbahire.com/10-framework-php-tot-nhat-cho-lap-trinh-vien) hay Perl…

### 2.1.2. Đặc điểm

* **Hệ quản trị cơ sở dữ liệu MySQL**chính là một trong những phần mềm quản trị CSDL dạng server based, hệ gần giống với [SQL server](https://www.microsoft.com/en-us/sql-server/sql-server-2017) of Microsoft.
* MySQL là phần mềm quản lý dữ liệu thông qua CSDL. Và mỗi một CSDL đều có bảng quan hệ chứa dữ liệu riêng biệt.
* MySQL có cơ chế quản lý sử dụng riêng giúp cho mỗi người sử dụng đều có thể quản lý cùng lúc một hay nhiều CSDL khác nhau. Và mỗi người dùng đều có 1 username và password để truy nhập và truy xuất đến CSDL. Khi truy vấn đến CSDL của MySQL, bạn phải cung cấp tài khoản và mật khẩu có quyền sử dụng cơ sở dữ liệu đó.
* MySQL tương thích tốt với môi trường Java, giúp hệ thống hoạt động mạnh mẽ. Bạn có thể viết hoặc chỉnh sửa code MySQL trên các IDE dành cho Java.

2.1.3. Lợi ích khi sự dụng MySQL



Hình 2.1: Lợi ích khi sử dụng MySQL

Ra mắt người dùng hoàn toàn miễn phí,**hệ quản trị cơ sở dữ liệu MySQL**mang đến rất nhiều những lợi ích thiết thực, được ứng dụng trong nhiều dự án của các công ty công nghệ, một trong số đó có thể kể đến những công ty chuyên lập trình như [MonaMedia](https://mona.media/), Misa,… cùng tìm hiểu những lợi ích thiết thực của mySQL ngay dưới đây:

* Hiệu năng sử dụng cao

Hầu hết, **hệ quản trị cơ sở dữ liệu MySQL** đang được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Và họ đánh giá rất cao ở hiệu năng sử dụng của MySQL. Với kiến trúc storage-engine, MySQL đặc trưng cho các ứng dụng chuyên biệt, đặc biệt là đối với những trang web có dung lượng lớn, phục vụ hàng triệu khách hàng. Hoặc đối với những hệ thống xử lý giao dịch tốc độ cao thì MySQL đều cùng có thể đáp ứng được những khả năng xử lý khắt khe của mọi hệ thống. Đặc biệt, với những tiện ích tải tốc độ cao, cơ chế xử lý nâng cao cùng bộ nhớ cache. MySQL đưa ra tất cả những tính năng cần có, đây là giải pháp hoàn hảo nhất ngay cả đối với những hệ thống doanh nghiệp khó tính nhất hiện nay.

* Hỗ trợ giao dịch mạnh mẽ

**Hệ quản trị cơ sở dữ liệu MySQL**không những mang lại hiệu năng sử dụng cao. Mà nó còn đưa ra một trong số những engine giao dịch cơ sở dữ liệu tốt nhất trên thị trường hiện nay. Tính năng này bao gồm: Khóa mức dòng không hạn chế; hỗ trợ giao dịch ACID hoàn thiện; khả năng giao dịch được phân loại và hỗ trợ giao dịch đa dạng mà người đọc không cản trở cho người viết và ngược lại. Với MySQL, dữ liệu sẽ được đảm bảo trong suốt quá trình server có hiệu lực. Các mức giao dịch độc lập sẽ được chuyên môn hóa, nếu phát hiện có lỗi khóa chết ngay tức thì.

* Tốc độ rất nhanh

Đánh giá chung của các nhà phát triển, tất cả họ đều cho rằng **hệ quản trị cơ sở dữ liệu MySQL**là cơ sở dữ liệu nhanh nhất. Đây là nơi để cho các website có thể trao đổi thường xuyên các dữ liệu bởi nó có engine xử lý tốc độ cao. Khả năng chèn dữ liệu cực nhanh và hỗ trợ mạnh mẽ các chức năng chuyên dụng cho trang web. Các tính năng này cũng được sử dụng cho môi trường lưu trữ dữ liệu mà hệ quản trị này tăng cường đến hàng terabyte cho các server đơn. Ngoài ra còn có những tính năng khác như: chỉ số băm, bảng nhớ chính, bảng lưu trữ và cây B được gói lại để giúp giảm các yêu cầu lưu trữ tối đa đến 80%. Với tốc độ nhanh, thật không thể phủ nhận **hệ quản trị cơ sở dữ liệu MySQL**là sự lựa chọn tốt nhất cho cả ứng dụng web cũng như các ứng dụng của doanh nghiệp ngày nay.

* Dễ dàng sử dụng

MySQL ngoài được biết đến với tốc độ khá cao, ổn định thì nó thực sự là một hệ thống cơ sở dữ liệu rất đơn giản, rất dễ sử dụng. Ít phức tạp khi cài đặt và quản trị hơn các hệ thống lớn. Đặc biệt nó có thể hoạt động trên tất cả các hệ điều hành.

* Hỗ trợ ngôn ngữ truy vấn

**MySQL hệ quản trị cơ sở dữ liệu**là ngôn ngữ của sự lựa chọn cho tất cả các hệ thống cơ sở dữ liệu hiện đại. Người dùng hoàn toàn có thể truy cập MySQL bằng cách sử dụng các ứng dụng mà hỗ trợ ODBC (một giao thức giao tiếp cơ sở dữ liệu được phát triển bởi Microsoft). Nhiều client có thể truy cập đến server trong cùng một thời gian. Đặc biệt các client có thể sử dụng nhiều cơ sở dữ liệu một cách đồng thời. Bạn cũng có thể truy cập MySQL tương tác với khi sử dụng một vài giao diện để đưa vào các truy vấn và xem kết quả như: các dòng yêu cầu của khách hàng, các trình duyệt web…

* Tính bảo mật và kết nối cao

Điều quan trọng nhất của mỗi một doanh nghiệp chính là việc bảo mật dữ liệu tuyệt đối. Và **hệ quản trị cơ sở dữ liệu MySQL**tích hợp các tính năng bảo mật an toàn tuyệt đối. MySQL được nối mạng một cách đầy đủ. Các cơ sở dữ liệu có thể được truy cập từ bất cứ nơi nào trên internet. Bạn có thể chia sẻ dữ liệu của bạn với bất kì ai, bất cứ lúc nào và bất cứ nơi đâu bạn muốn. Nhưng MySQL kiểm soát quyền truy cập nên người không nên nhìn thấy dữ liệu của bạn sẽ không thể nào nhìn được. Với việc xác nhận truy cập cơ sở dữ liệu, MySQL trang bị các kĩ thuật mạnh. Chỉ có những người sử dụng đã được xác nhận mới truy cập được vào cơ sở dữ liệu. Ngoài ra, SSH và SSL cũng được hỗ trợ nhằm đảm bảo kết nối an toàn và bảo mật. Tiện ích backup và recovery cung cấp bởi **MySQL hệ quản trị cơ sở dữ liệu**và các hãng phần mềm thứ 3 cho phép backup logic và vật lý cũng như recovery toàn bộ ngay tại một thời điểm.

* Tính linh động cao

**MySQL** có thể hoạt động trên tất cả các hệ điều hành, chạy được với mọi phần cứng từ các máy PC ở nhà cho đến các máy server. Máy chủ **hệ quản trị cơ sở dữ liệu MySQL**đáp ứng nhiều tính năng linh hoạt. Nó có sức chứa để xử lý các ứng dụng được nhúng sâu với 1MB dung lượng để chạy kho dữ liệu khổng lồ lên đến hàng terabytes thông tin. Tính chất mã nguồn mở của MySQL cho phép tùy biến theo ý muốn để thêm các yêu cầu phù hợp cho database server.

* Mã nguồn mở tự do

Băn khoăn của rất nhiều doanh nghiệp khi họ gặp khó khăn trong việc giao toàn bộ phần mềm cho một mã nguồn mở. Bởi khó có thể tìm được hỗ trợ hay bảo mật an toàn phục vụ một cách chuyên nghiệp. Nhưng vấn đề khó khăn này sẽ không còn nữa nếu sử dụng **hệ quản trị cơ sở dữ liệu MySQL.**Với MySQL, mọi sự cam kết đều rất rõ ràng, mọi sự cố đều được MySQL bồi thường. Bạn có thể tìm thấy các tài nguyên có sẵn mà hệ quản trị này hỗ trợ. Cộng đồng MySQL rất có trách nhiệm. Họ thường trả lời các câu hỏi trên mailing list chỉ trong vài phút. Nếu lỗi xảy ra, các nhà phát triển sẽ đưa ra cách khắc phục nhanh nhất cho bạn. Và cách khắc phục đó sẽ ngay lập tức có sẵn trên internet.

* Chi phí sở hữu thấp

**Hệ quản trị cơ sở dữ liệu MySQL**cung cấp miễn phí cho hầu hết các việc sử dụng trong một tổ chức. Chính vì vậy, sử dụng MySQL cho các dự án, các doanh nghiệp đầu nhận thấy được sự tiết kiệm cho phí rất đáng kể. Người dùng của MySQL cũng không phải mất nhiều thời gian để sửa chữa hoặc vấn đề thời gian chết.

Không gì hoàn hảo hơn khi doanh nghiệp của bạn có được sự hỗ trợ đắc lực từ **hệ quản trị cơ sở dữ liệu MySQL**. Cơ hội phát triển sẽ nằm ngay trong tầm tay thật đơn giản, dễ dàng, hiệu quả cao.



## **2.2. Tổng quan về Java**

### 2.2.1. Khái niệm

Java là một ngôn ngữ lập lập trình, được phát triển bởi **Sun Microsystem** vào năm 1995, là ngôn ngữ kế thừa trực tiếp từ C/C++ và là một ngôn ngữ lập trình hướng đối tượng.

Java là tên một hòn đảo ở Indonesia - hòn đảo nổi tiếng với loại coffee Peet và cũng là loại nước uống phổ biến của các kỹ sư Sun. Ban đầu Ngôn ngữ này được đặt tên là "Oak" (có nghĩa là "Cây sồi" - 1991), nhưng các luật sư của Sun xác định rằng tên đó đã được đăng ký nhãn hiệu nên các nhà phát triển đã phải thay thế bằng một tên mới -  và cũng vì lý do trên mà cái tên Java đã ra đời và trở thành tên gọi chính thức của Ngôn ngữ này - Ngôn ngữ Lập trình Java.

* Ứng dụng của Java
* Phát triển ứng dụng cho các thiết bị điện tử thông minh, các ứng dụng cho doanh nghiệp với quy mô lớn.
* Tạo các trang web có nội dung động *(web applet*), nâng cao chức năng của server.
* Phát triển nhiều loại ứng dụng khác nhau: Cơ sở dữ liệu, mạng, Internet, viễn thông, giải trí,...

### 2.2.2.. Đặc điểm của Java

Tiêu chí hàng đầu của Ngôn ngữ Lập trình Java là **"**Write Once, Run Anywhere**"** (*Viết một lần, chạy mọi nơi*), nghĩa là Java cho phép chúng ta viết code một lần và thực thi được trên các hệ điều hành khác nhau. Ví dụ, viết code trên Hệ điều hành Windows và nó có thể thực thi được trên các Hệ điều hành Linux và Mac OS...

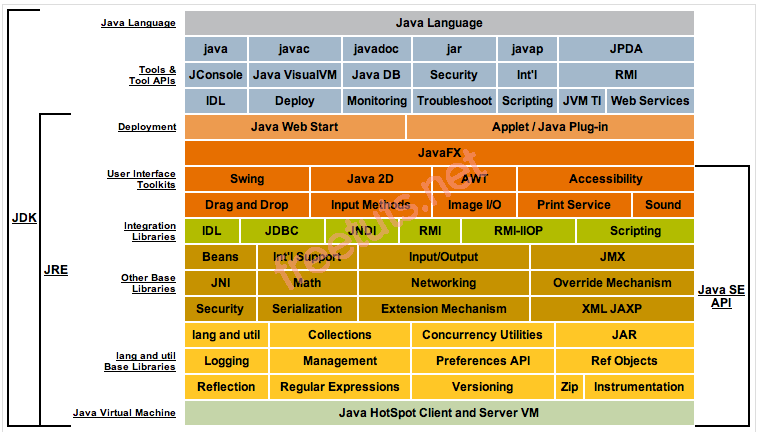
Với đặc điểm nổi bật đó, Java có những đặc điểm cơ bản như sau:

* Đơn giản và quen thuộc: Vì Java kế thừa trực tiếp từ C/C++ nên nó có những đặc điểm của ngôn ngữ này, Java đơn giản vì mặc dù dựa trên cơ sở C++ nhưng Sun đã cẩn thận lược bỏ các tính năng khó nhất của của C++ để làm cho ngôn ngữ này dễ sử dụng hơn.
* Hướng đối tượng và quen thuộc.
* Mạnh mẽ *(thể hiện ở cơ chế tự động thu gom rác - Garbage Collection*) và an toàn.
* Kiến trúc trung lập, độc lập nền tảng và có tính khả chuyển (*Portability*).
* Hiệu suất cao.
* Máy ảo (biên dịch và thông dịch).
* Phân tán.
* Đa nhiệm: Ngôn ngữ Java cho phép xâ dựng trình ứng dụng, trong đó nhiều quá trình có thể xảy ra đồng thời. Tính đa nhiệm cho phép các nhà lập trình có thể biên soạn phần mềm đáp ứng tốt hơn, tương tác tốt hơn và thực hiện theo thời gian thực.

### 2.2.3. Các platform của Java

Java Platform gồm có 3 thành phần chính:

* Java Virtual Machine (*Java VM*): Máy ảo Java.
* Java Application Programming Interface (*Java API*).
* Java Development Kit (*JDK*) gồm trình biên dịch, thông dịch, trợ giúp, soạn tài liệu... và các thư viện chuẩn.

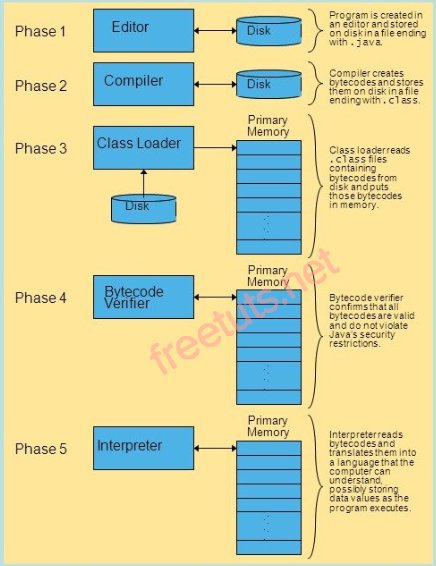


Hình 2.2: Platform cơ bản của Java

### 2.2.4. Tiêu chuẩn của một môi trường Java điển hình

Thông thường, các chương trình Java trải qua 5 giai đoạn chính:

* Editor: Lập trình viên viết chương trình và được lưu vào máy tính với định dạng .java.
* Compiler: Biên dịch chương trình thành bytecodes (*định dạng .class*) - nhờ bước trung gian này mà Java được viết 1 lần và chạy trên các hệ điều hành khác nhau.
* Class Loader: Đọc file .class chứa mã bytecodes và lưu vào trong bộ nhớ.
* Bytecode Verifier: Đảm bảo rằng mã bytecodes là hợp lệ và không vi phạm các vấn đề về bảo mật của Java.
* Intepreter: Biên dịch bytecodes thành mã máy để máy tính có thể hiểu được và sau đó thực thi chương trình.



Hình 2.3: Môi trường java điển hình

## 2.3. Tổng quan về ReactJS

ReactJS là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook, ra mắt vào năm 2013 với mục đích để xây dựng giao diện người dùng. Nó được sử dụng rộng rãi để xây dựng các trang web SPA (Single Page Application) và các ứng dụng trên nền tảng di động. Nó rất dễ sử dụng và cho phép người dùng có thể tạo các component UI có thể tái sử dụng.

* Những tính năng của ReactJS

ReactJS có các tính năng hết sức nổi bật bao gồm:

* JSX: viết tắt của JavaScript extension, nó là React extension , giúp cho việc thay đổi cây DOM dễ dàng hơn bằng HTML-style code đơn giản. Nó là một trong những tính năng tốt và dễ sử dụng.
* Components: Một trang web được xây dựng bằng ReactJS là một sự kết hợp nhiều component lại với nhau chứ không phải chung một Template như bình thường. Các component cũng như các hàm JavaScript bình thường, giúp tạo ra các code dễ dàng bằng cách tách các logic ra thành các đoạn code độc lập có thể tái sử dụng. Chúng ta có thể sử dụng component dưới dạng function hoặc class, ngoài ra các component còn có state và props.
* Virtual DOM: ReactJS tạo một thứ gọi là Virtual DOM (DOM ảo). Đúng như tên gọi, nó là một copy của DOM thật trên trang web đó. ReactJS dùng những DOM ảo đó để tìm đúng những DOM thật cần được cập nhật khi có bất kỳ sự kiện nào làm các thành phần bên trong nó thay đổi.
* Javascript Expressions: Biểu thức JS có thể sử dụng trong file .jsx hoặc .js bằng cách sử dụng cặp dấu ngoặc nhọn “{}”.

Ưu và nhược điểm.

Mỗi framework hay thư viện nào cũng có ưu và nhược điểm cả, ReactJS cũng vậy. Dưới đây là ưu và nhược điểm của ReactJS.

* Ưu điểm của ReactJS:

Vì ReactJS sử dụng DOM ảo để cache cấu trúc dữ liệu trong bộ nhớ và chỉ những thay đổi cuối cùng mới được cập nhật vào trong DOM trình duyệt. Điều này làm cho ứng dụng trở nên nhanh hơn.

Bạn có thể tạo các component theo từng chức năng mà bạn muốn bằng cách sử dụng tính năng react component. Các component này có thể tái sử dụng theo nhu cầu của bạn, đồng thời việc tạo các component theo từng chức năng cũng giúp cho việc bảo trì sau này trở nên dễ dàng hơn.

ReactJS là một opensource, vì vậy cũng rất dễ cho những bạn mới bắt đầu tìm hiểu nó.

Trong những năm gần đây, ReactJS đang trở nên phổ biến hơn và được duy trì bởi Facebook và Instagram. Ngoài ra nó cũng được sử dụng bởi các công ty nổi tiếng như Apple, Netflix, …

Facebook vẫn đang duy trì, phát triển, và cho ra những thay đổi mới. Vì thế bạn cứ yên tâm sử dụng ReactJS cho những dự án của bạn hoặc những dự án dành do doanh nghiệp.

ReactJS có thể được sử dụng để xây dựng giao diện người dùng cho cả các ứng dụng dành cho máy tính và các ứng dụng di động.

Dễ dàng cho việc test và debug, vì hầu hết các code đều được thực hiện bằng JavaScript chứ không phải bằng HTML.

* Nhược điểm của ReactJS

Vì hầu hết code được viết dưới dạng JSX, tức là HTML và CSS là một phần của JavaScript, nó không giống như những framework khác vẫn tách biệt giữa HTML và CSS nên những bạn mới làm quen với ReactJS sẽ hơi lúng túng và dễ nhầm lẫn giữa JSX và HTML. Tuy nhiên bạn sẽ nhanh chóng quen với cách kết hợp này của React mà thôi.

Một nhược điểm nữa của ReactJS đó là dung lượng các file của nó hơi lớn.



## **2.4. Tổng quan về StarUML**

* StarUML là môi trường mô hình hóa tổng thể dưới dạng đồ họa và dễ dàng sử dụng. Công dụng StarUML cung cấp:
* Trình bày các mô hình bằng các ký hiệu đồ họa phù hợp với các phương pháp mô hình hóa trong thực tế như mô hình thực thể và mối kết hợp, mô hình quan hệ...
* Tự động tạo hồ sơ mô tả các đối tượng trên mô hình.
* Tự động tạo mã phát sinh CSDL và các chức năng xử lý từ mô hình đã xây dựng.
* Vai trò của StarUML trong việc thiết kế mô hình cơ sở dữ liệu:
* Trình bày mô hình ở dạng đồ họa.
* Kiểm tra tính hợp lệ của mô hình được thiết kế
* Phát sinh mô hình dữ liệu logic hay mô hình dữ liệu vật lý.
* Sơ lược về thực thể, thuộc tính, liên kết

Thực thể

* Thực thể là khái niệm để chỉ một lớp các đối tượng có cùng đặc tính chung mà người ta muốn quản lý thông tin về nó. Ví dụ, sinh viên, hàng hóa, vật tư…
* Một đối tượng cụ thể trong thực thể được gọi là một cá thể (còn gọi là một thể hiện của thực thể). Ví dụ Lê Xuân Hưng là một cá thể của thực thể Sinh viên. Bánh kẹo là các thể của thực thế hàng hóa.

Thuộc tính thực thể

* Để mô tả thông tin về một thực thể người ta thường dựa vào các đặc trưng riêng của thực thể đó, Các đặc trưng đó được gọi là thuộc tính của thực thể. Ví dụ thực thể Sinh viên có các thuộc tính Mã sinh viên, Họ tên sinh viên, Ngày sinh, Địa chỉ, Trường, Khoa, Khóa sinh viên, Lớp,...

**Liên kết**

* Một liên kết là một sự ghép nối giữa hai hay nhiều thực thể. Phản ánh sự liên hệ giữa các thực thể Có 3 kiểu liên kết: liên hết một một (1-1), quan hệ nhiều nhiều (n-n) và quan hệ một nhiều (1-n).
* Quan hệ 1-1: Hai thực thể A và B có mối quan hệ 1-1 nếu một thực thể kiểu A tương ứng với một thực thể kiểu B và ngược lại.
* Quan hệ 1-n: Hai thực thể A và B có mối quan hệ 1- n nếu một thực thể kiểu A tương ứng với nhiều thực thể kiểu B và một thực thể của B chỉ tương ứng với một thực thể kiểu A.
* Quan hệ n-n: Hai thực thể A và B có mối quan hệ n-n nếu một thực thể kiểu A tương ứng với nhiều thực thể kiểu B và ngược lại.

# CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## **3.1. Các bước hoàn thành dự án**

*Bảng 3-1: Các bước hoàn thành dự án*

|  |  |
| --- | --- |
| **Tiến trình** | **Mô tả** |
| Khởi tạo dự án | Tìm hiểu các tài liệu liên quan đến dự án.  Tìm hiểu cách thức hoạt động của việc bán hàng và giao  hàng trên mạng.  Xem xét tính khả thi của dự án, các khía cạnh liên quan  đến dự án. |
| Lập kế hoạch | Mô tả tổng quát quá trình thực hiện dự án.  Ràng buộc dự án với các điều kiện chủ quan và khách  quan. Xác định tính khả thi của dự án.  Xác định rủi ro và các vấn đề phát sinh trong quá trình  xây dựng dự án có thể gặp.  Lập kế hoạch hệ thống mới. |
| Thu thập yêu cầu | Thu thập yêu cầu của khách hàng đối với hệ thống một  cách chi tiết và chính xác.  Khảo sát, tìm hiểu các hệ thống đã có thể đưa ra các yêu  cầu, các chức năng cho hệ thống của mình. |
| Phân tích và kiểm tra  yêu cầu | Phân tích các yêu cầu chức năng và phi chức năng.  Kiểm tra tính khả thi của yêu cầu.  Bổ sung các yêu cầu còn thiếu sót. |
| Phân tích | Mô hình hóa các yêu cầu hệ thống và đặc tả các chức  năng. |
| Thiết kế | Thiết kế cơ sở dữ liệu. Thiết kế giao diện. |
| Kiểm thử | Kiểm thử cho từng giai đoạn của dự án từ lúc bắt đầu đến lúc kết thúc dự án. |
| Cài đặt | Xây dựng website cho kế hoạch đã đặt ra.  Viết tài liệu hướng dẫn sử dụng hệ thống người dùng. |
| Thực thi | Đưa hệ thống vào hoạt động thử. Theo dõi hoạt động của hệ thống. |

**3.2. Phân tích thiết kế hệ thống**

### 3.2.1. Tác nhân và ca sử dụng

* Khách hàng: Là người có nhu cầu gửi xe, sử dụng dịch vụ bãi đỗ xe.
* Người quản trị: Là người điều hành các bãi đỗ xe, khách hàng.

### 3.2.2. Use case

* Quản lý bãi đỗ xe ( thêm, sửa, xóa, tìm kiếm)
* Quản lý hợp các khối trong bãi ( thêm, sửa, xóa, tìm kiếm )
* Quản lý các vị chỗ đỗ ( thêm, sửa, xóa, tìm kiếm )
* Quản lý khách hàng ( thêm, sửa, xóa, tìm kiếm )
* Quản lý tài khoản ( đăng nhập, đăng kí, đổi mật khẩu, khóa, phân quyền, tìm kiếm )
* Thuê dịch vụ (đặt lịch, sửa, xóa, xem lịch sử, lọc)

### Đặc tả ca sử dụng

#### Quản lý bãi đỗ xe

Ca sử dụng dùng để quản lý bãi đỗ xe

* Admin đăng nhập vào hệ thống bằng tài khoản của mình, lựa chọn từng chức năng phù hợp với yêu cầu: thêm/sửa/xóa.
* Nếu lựa chọn, thêm hoặc sửa thông tin bãi đỗ. Sau khi nhập và điều chỉnh đầy đủ thông tin cần điều chỉnh hoặc thêm, admin click vào cập nhật thông tin về bãi tự động thêm vào CSDL.
* Nếu admin lựa chọn xóa thì click vào icon thùng rác. Sau đó, nó sẽ xóa bãi đỗ xe đó

#### Quản lý các khối trong bãi

Ca sử dụng để quản khối của admin.

* Admin đăng nhập vào hệ thống bằng tài khoản của mình, lựa chọn từng chức năng phù hợp với yêu cầu: thêm/sửa/xóa.
* Admin có thể thêm sửa thông tin các khối. Sau khi nhập và điều chỉnh đầy đủ thông tin cần điều chỉnh hoặc thêm, admin click vào cập nhật thông tin.
* Nếu admin lựa chọn xóa thì click vào icon xóa. Sau khi ấn, thông tin của khối đó sẽ được xóa khỏi CSDL.

#### Quản lý các vị trí đỗ

Ca sử dụng để quản lý vị trí đỗ của admin.

* Admin đăng nhập vào hệ thống bằng tài khoản của mình, lựa chọn từng chức năng phù hợp với yêu cầu: thêm/sửa/xóa.
* Admin có thể thêm sửa thông tin các khối. Sau khi nhập và điều chỉnh đầy đủ thông tin cần điều chỉnh hoặc thêm, admin click vào cập nhật thông tin.
* Nếu admin lựa chọn xóa thì hãy điền số vị trí muốn xóa. Sau khi xác nhận, thông tin của số vị trí đỗ sẽ được xóa khỏi CSDL theo số từ lớn nhất.

#### Quản lý khách hàng

* Khách hàng tạo tài khoản sẽ được lưu vào CSDL. Chia ra làm thông tin đăng nhập và thông tin cá nhân
* Admin có thể thực hiện cấm hoặc xóa tài khoản khách hàng.

#### Quản lý tài khoản

Ca sử dụng dùng để người quản trị quản lý khách hàng.

* Ca sử dụng được dùng khi kiểm tra tài khoản và phân quyền cho tài khoản.
* Người quản trị đăng nhập vào hệ thống bằng tài khoản của mình, với quyền quản trị viên lựa chọn chức năng phù hợp với yêu cầu: khóa/ tìm kiếm
* Nếu khách hàng vi phạm các điều khoản của công ty đưa ra tài khoản đó có thể bị khóa và không thể sử dụng được nữa.

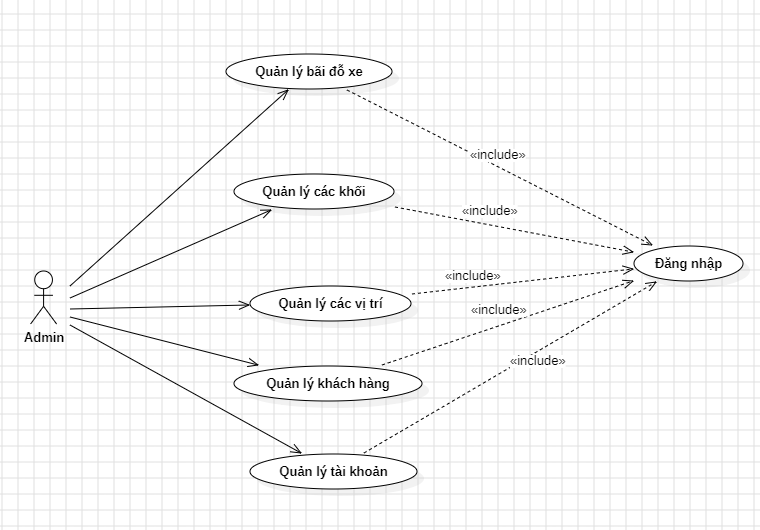
Ca sử dụng là khách hàng.

* Ca sử dụng được dùng khi kiểm tra tài khoản và phân quyền cho tài khoản.
* Khách hàng đăng nhập vào hệ thống, có thể lựa chọn chức năng phù hợp với yêu cầu: xem/ cập nhật/ vé tháng.

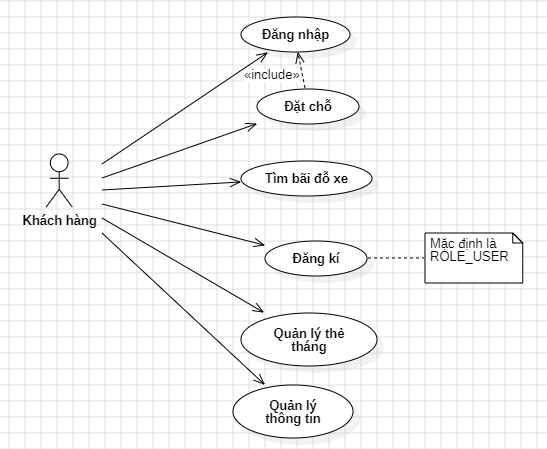
#### Thuê dịch vụ

* Khách hàng có thể xem các lịch đặt, lịch sử đặt lịch cũng như thêm lịch, sửa lịch, xóa lịch.

## **Biểu đồ ca sử dụng tham gia quản lý bãi đỗ xe**

****

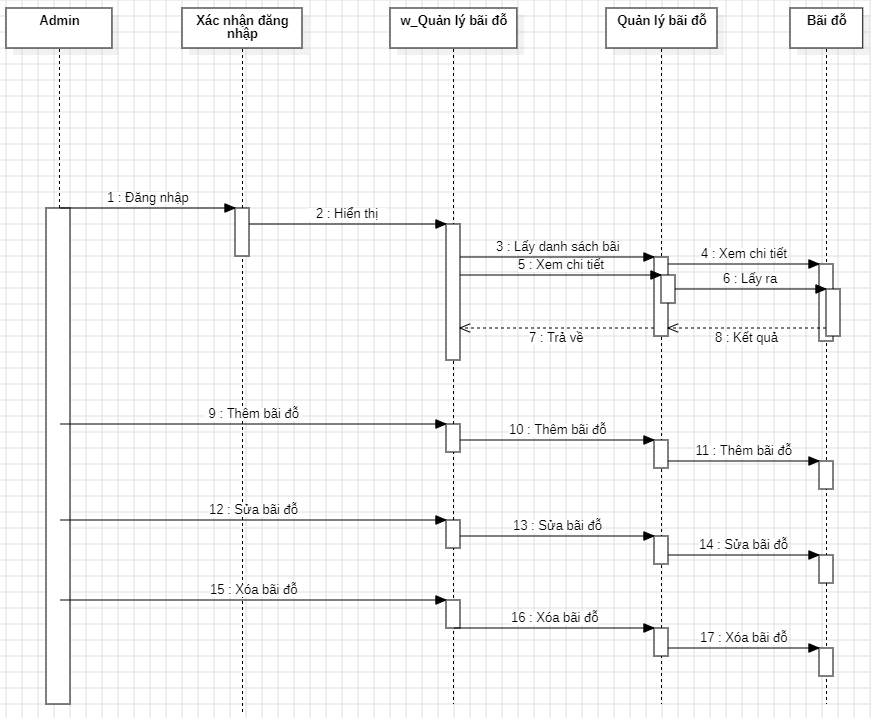
Hình 3.1. Biểu đồ ca sử dụng của admin

**

*Hình 3.2: Biểu đồ ca sử dụng của khách hàng*

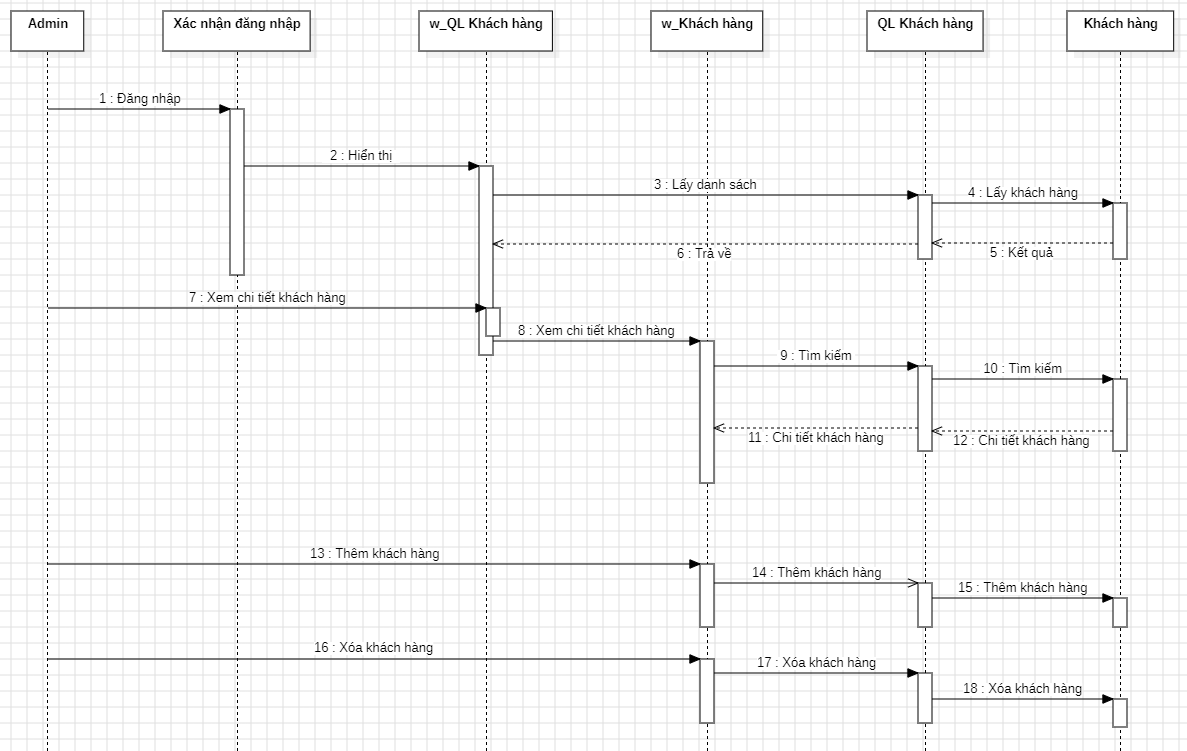
## **Biểu đồ trình tự**

### 3.4.1. Quản lý bãi đỗ

**`**

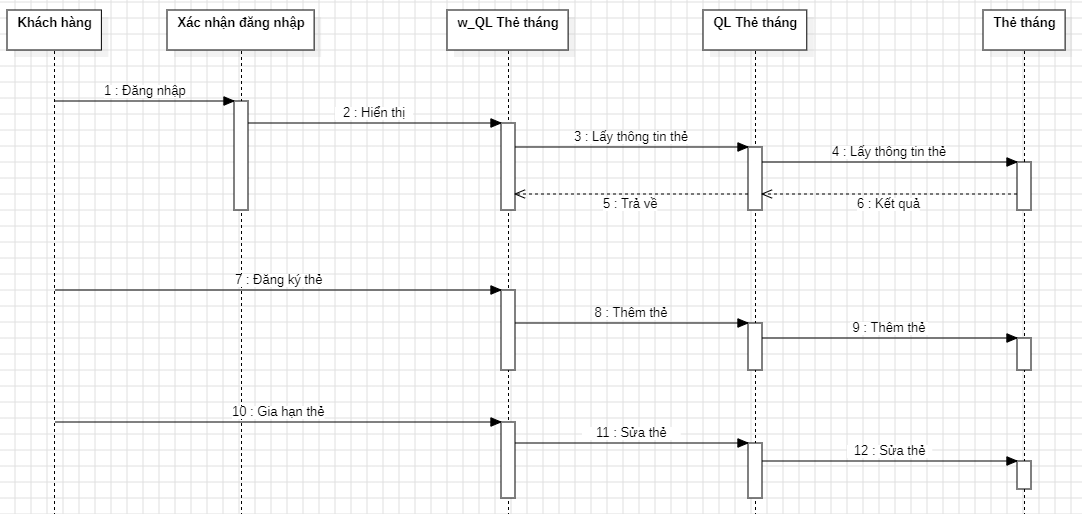
Hình 3.6: Biểu đồ trình tự bãi đỗ

### 3.4.2. Quản lý khách hàng

****

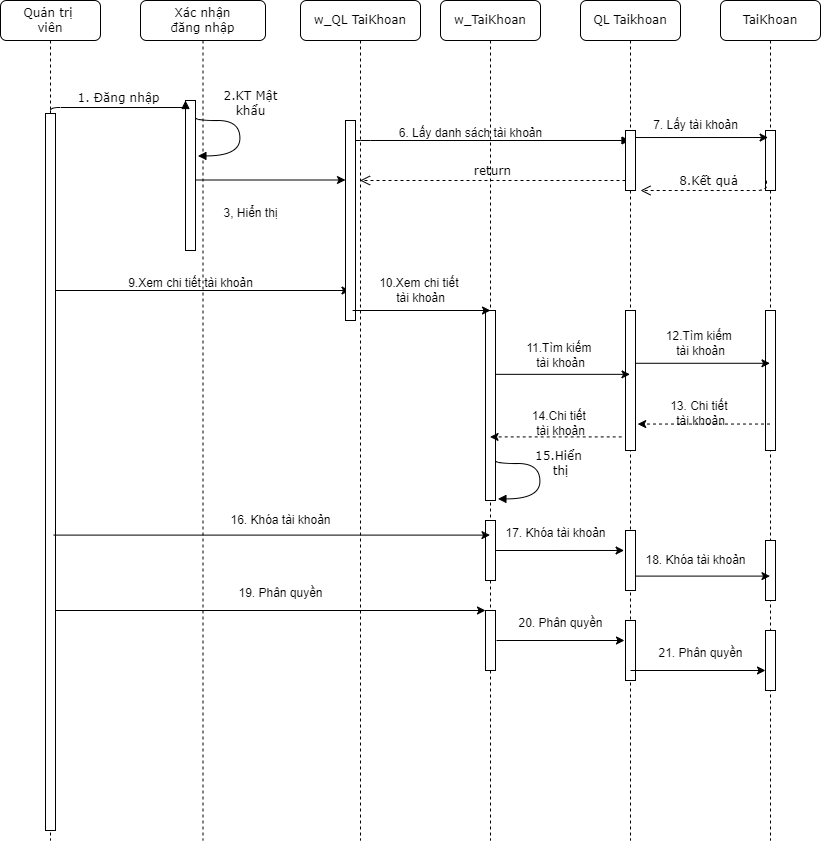
Hình 3.7: Biểu đồ trình tự khách hàng

### Quản lý thẻ tháng

****

Hình 3.8: Biểu đồ trình tự thẻ tháng

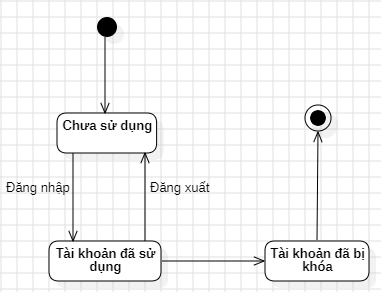
### Quản lý tài khoản

****

Hình 3.11: Biểu đồ trình tự quản lý tài khoản

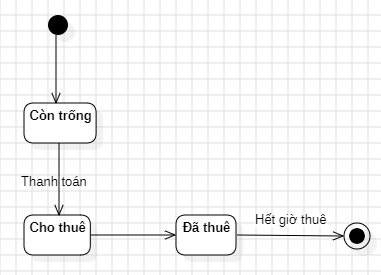
## **Biểu đồ trạng thái**

### 3.5.1. Tài khoản

****

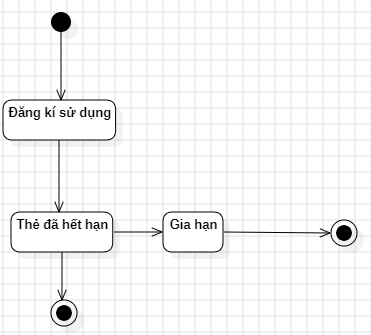
Hình 3.13. Biểu đồ trạng thái của tài khoản

### Vị trí đặt (Parking slot)

****

Hình 3.14: Biểu đồ trạng thái vị trí đặt

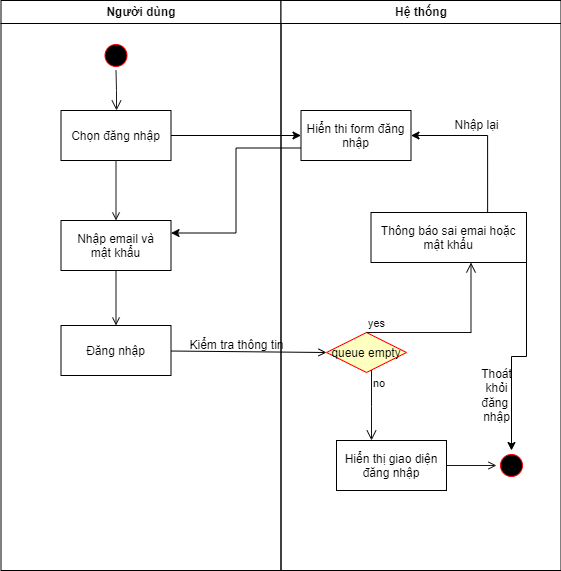
### Thẻ tháng (Regular pass)

****

Hình 3.15: Biểu đồ trạng thái của thẻ tháng

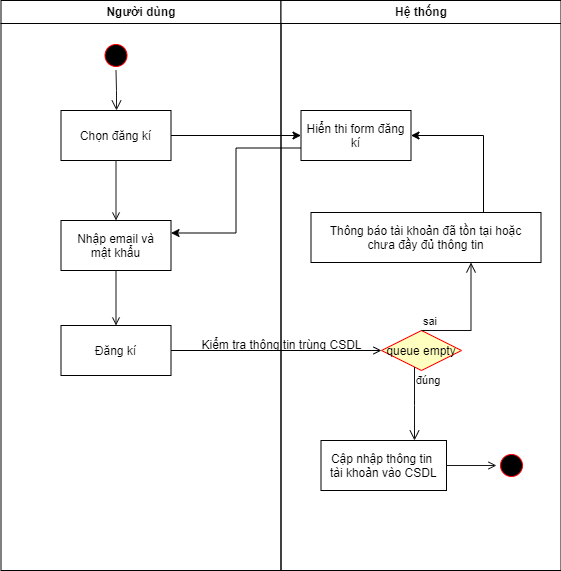
## **Biểu đồ hoạt động**

### 3.6.1. Chức năng đăng nhập

****

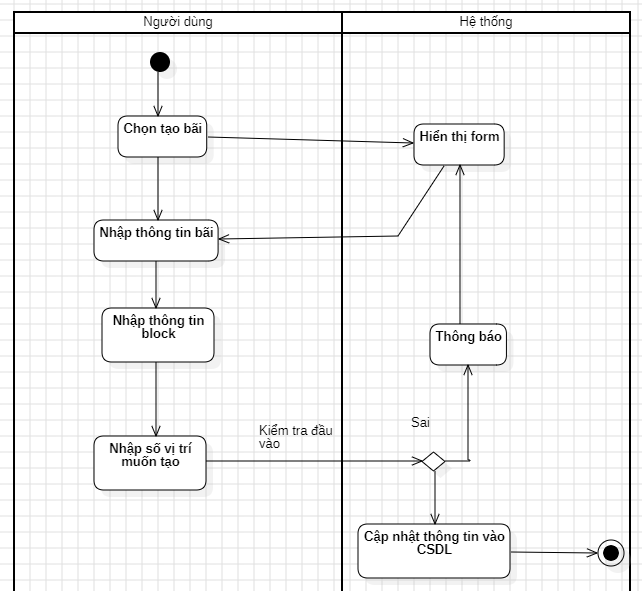
Hình 3.17: Biểu đồ hoạt động của đăng nhập

### Chức năng đăng kí

****

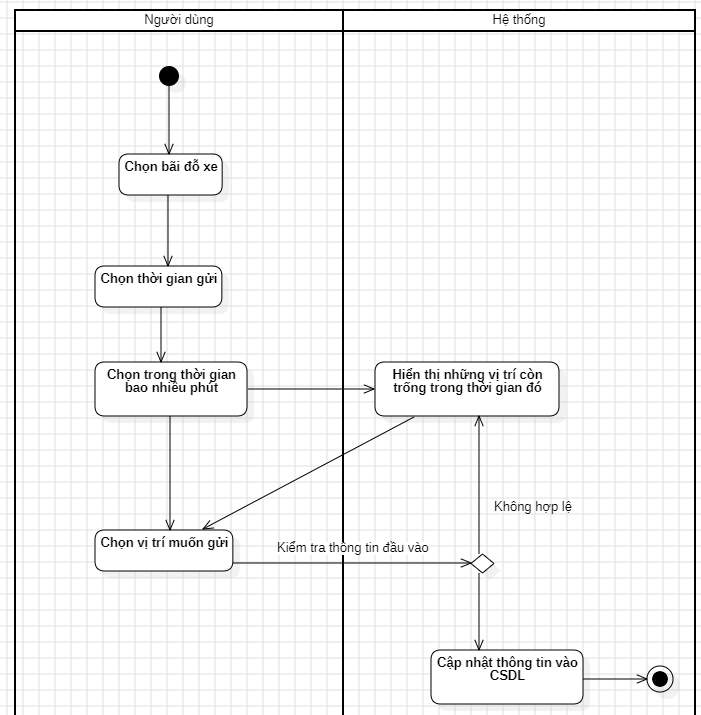
Hình 3.18: Biểu đồ hoạt động của đăng kí

### Chức năng tạo bãi đỗ xe

****

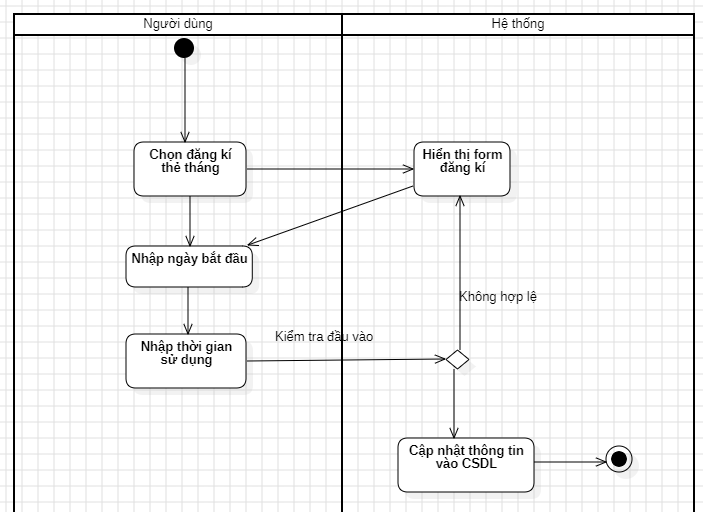
Hình 3.19: Biểu đồ hoạt động tạo bãi đỗ xe

### Chức năng đặt chỗ



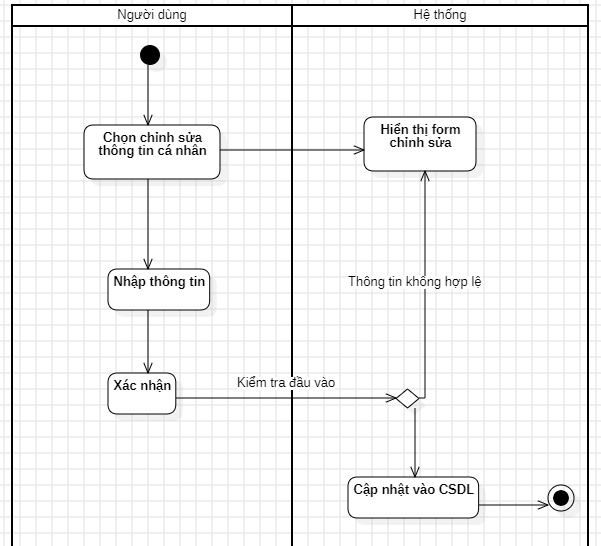
Hình 3.20: Biểu đồ hoạt động của chức năng đặt chỗ

### Chức năng đăng kí thẻ tháng



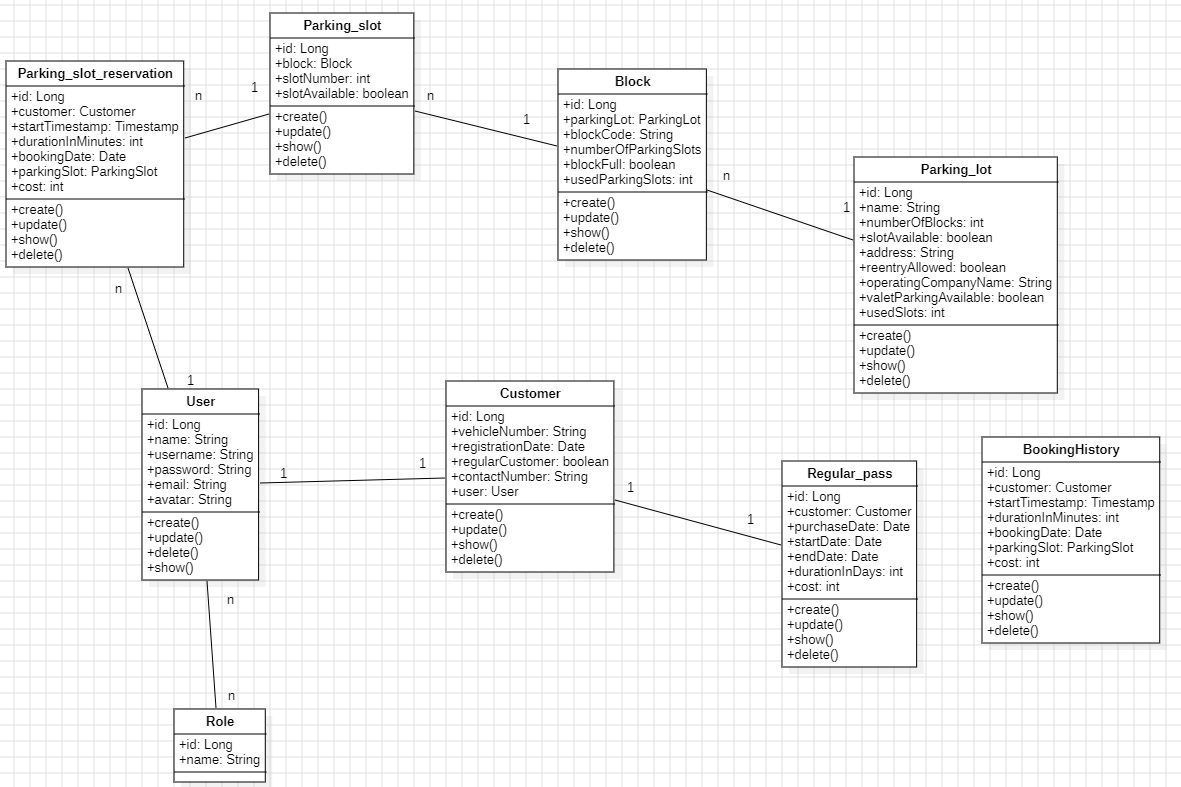
Hình 3.21: Biểu đồ hoạt động của đăng kí thẻ tháng

### Chức năng cập nhật thông tin cá nhân

****

Hình 3.22: Biểu đồ hoạt động của chức năng cập nhật thông tin

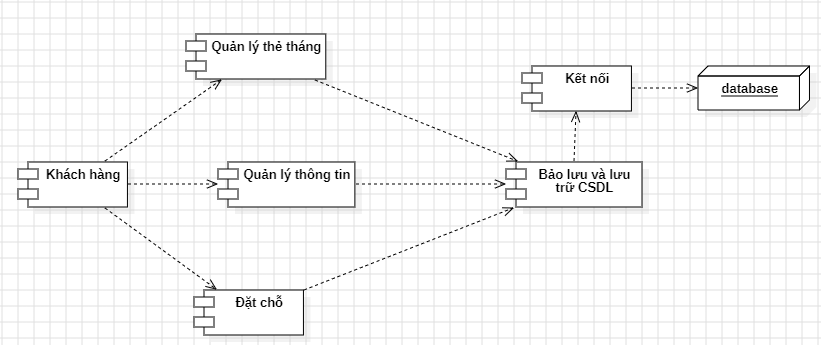
## **Biểu đồ lớp chi tiết**

****

Hình 3.28: Biều đồ lớp chi tiết

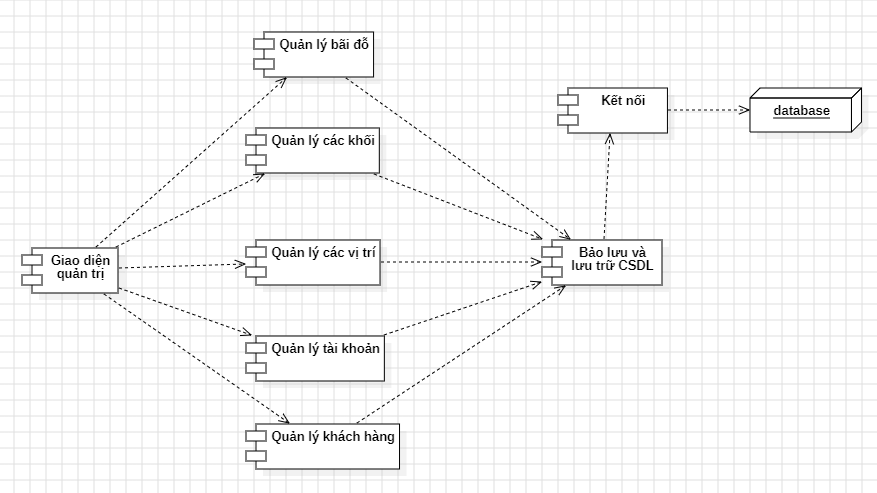
## **Biểu đồ thành phần**

### 3.8.1. Khách hàng

****

Hình 3.29: Biểu đồ thành phần người quản trị

### Admin

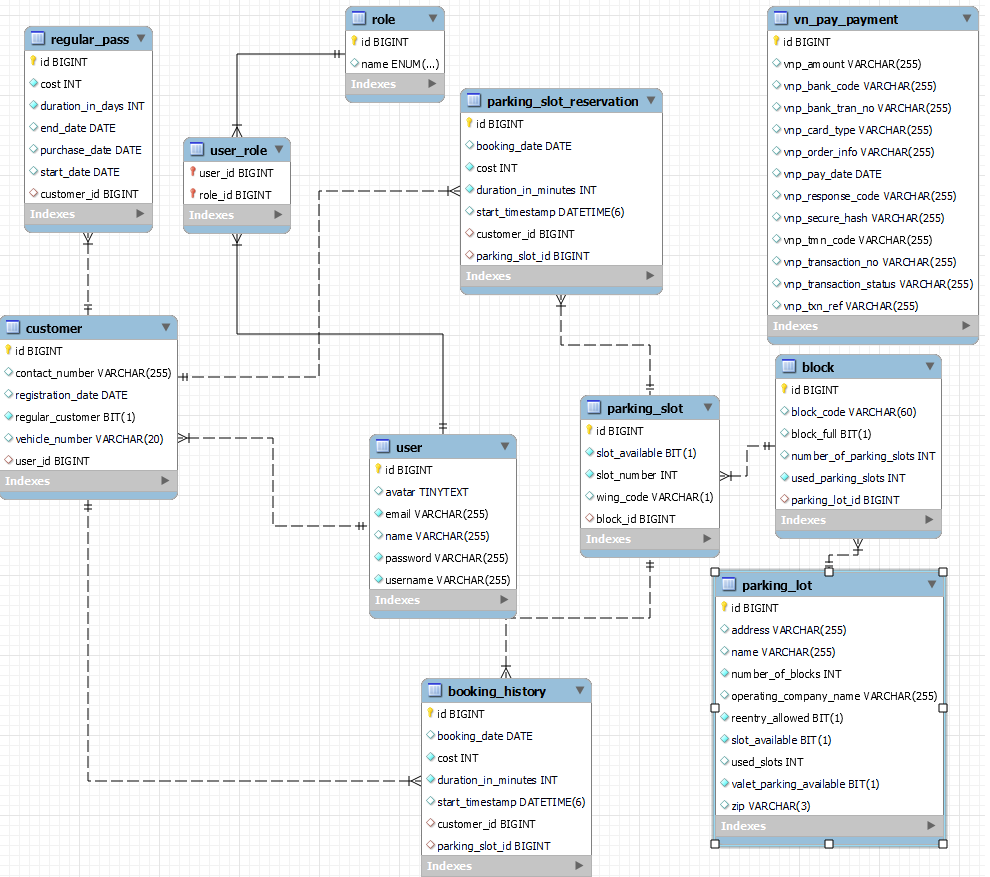
****

**

Hình 3.30: Biểu đồ thành phần admin

## **Thiết kế cơ sở dữ liệu**

### 3.9.1. Cơ sở dữ liệu và mối quan hệ

****

Hình 3.31: Cơ sở dữ liệu và mối quan hệ

### Mô tả chi tiết các bảng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Kích cỡ** | **Mô tả** | **Ghi chú** |
| Id | Bigint |  | Mã id  Người dùng | Khóa chính |
| avatar | TINYTEXT |  | Hình ảnh |  |
| Email | Varchar | 255 |  |  |
| Name | Varchar | 255 | Tên  Người dùng |  |
| Password | Varchar | 255 | Mật khẩu |  |
| username | Varchar | 255 | Tài khoản |  |

*Hình 3.2: Bảng User*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Kích cỡ** | **Mô tả** | **Ghi chú** |
| Id | bigint |  | Mã id của  Quyền | Khóa chính |
| Name | Varchar | 60 | Tên quyền |  |

*Bảng 3.3: Bảng Role*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Kích cỡ** | **Mô tả** | **Ghi chú** |
| Id | Bigint |  | Mã id của  phòng và bài đăng | Khóa chính |
| Address | Varchar | 255 | Địa chỉ |  |
| name | Varchar | 255 | Tên bãi |  |
| number\_of\_blocks | int |  |  |  |
| operating\_company\_name | Varchar | 255 | Tên công ty |  |
| reentry\_allowed | BIT | 1 | Cho phép quay lại |  |
| slot\_available | BIT | 1 | Còn trống |  |
| used\_slots | int |  | Đã dung bao nhiêu vị trí |  |
| valet\_parking\_available | BIT |  | Không đặt trước |  |

*Bảng 3.4: Bảng Parking\_lot*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Kích cỡ** | **Mô tả** | **Ghi chú** |
| Id | Bigint |  | Mã id của đánh giá | Khóa chính |
| block\_code | Varchar | 60 | Mã block |  |
| block\_full | BIT |  | Kiểm tra đầy |  |
| number\_of\_parking\_slots | int |  | Số vị trí của bãi |  |
| used\_parking\_slots | int |  | Số vị trí đã dùng |  |

*Bảng 3.5: Bảng Block*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Kích cỡ** | **Mô tả** | **Ghi chú** |
| Id | Bigint |  | Mã id của parking slot | Khóa chính |
| slot\_available | BIT |  | Kiểm tra vị trí khả dụng |  |
| slot\_number | int |  | Số định danh |  |
| block\_id | Bigint |  | Thời gian tạo | Khóa phụ |

*Bảng 3.6: Bảng Parking\_slot*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Kích cỡ** | **Mô tả** | **Ghi chú** |
| Id | Bigint |  | Mã id của lưu bài đăng | Khóa chính |
| contact\_number | Varchar | 20 | Số điện thoại khách hàng |  |
| registration\_date | DATE |  | Ngày tạo |  |
| regular\_customer | BIT |  | Có phải khách mua thẻ tháng |  |
| vehicle\_number  user\_id | Varchar  BIGINT | 20 | Biển số xe  Mã tài khoản | Khóa phụ |

*Bảng 3.7: Bảng Customer*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Kích cỡ** | **Mô tả** | **Ghi chú** |
| Id | Bigint |  | Mã id | Khóa chính |
| cost | int |  | Giá |  |
| start\_date | DATE |  | Ngày bắt đầu |  |
| purchase\_date | DATE |  | Ngày thanh toán |  |
| duration\_in\_days | int |  | Trong bao nhiêu ngày |  |
| customer\_id | BIGINT |  | Mã khách hàng | Khóa phụ |

*Bảng 3.8: Bảng regular\_pass*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Kích cỡ** | **Mô tả** | **Ghi chú** |
| Id | Bigint |  | Mã id của message chat | Khóa chính |
| booking\_date | DATE |  | Ngày đặt |  |
| cost | int |  | Phí gửi |  |
| duration\_in\_minutes | int | 1 | Thời gian gửi  theo phút |  |
| start\_timestamp | Datetime |  | Thời gian gửi |  |
| customer\_id | Bigint |  | Mã khách hàng | Khóa phụ |
| parking\_slot\_id | Bigint |  | Mã vị trí | Khóa phụ |

*Bảng 3.9: Bảng parking\_slot\_reservation*

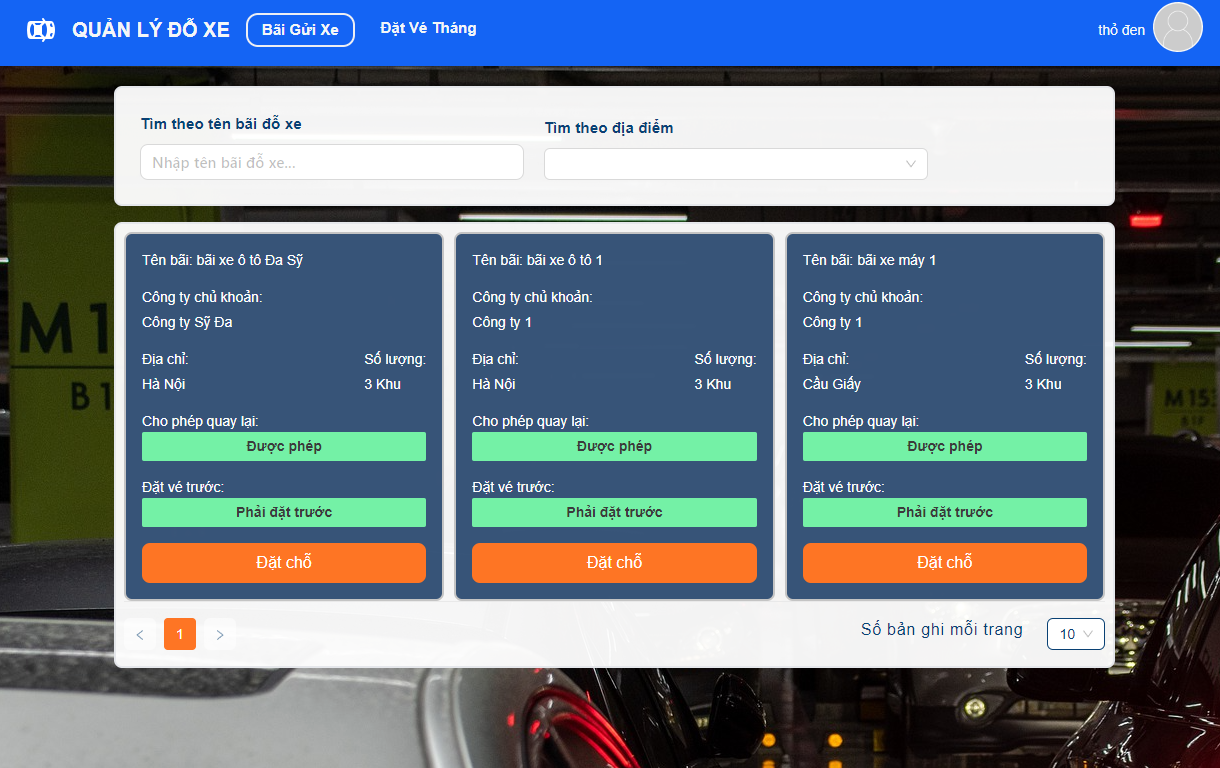
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Kích cỡ** | **Mô tả** | **Ghi chú** |
| Id | Bigint |  | Mã id của message chat | Khóa chính |
| booking\_date | DATE |  | Ngày đặt |  |
| cost | int |  | Phí gửi |  |
| duration\_in\_minutes | int | 1 | Thời gian gửi  Theo phút |  |
| start\_timestamp | Datetime |  | Thời gian gửi |  |
| customer\_id | Bigint |  | Mã khách hàng | Khóa phụ |
| parking\_slot\_id | Bigint |  | Mã vị trí | Khóa phụ |

*Bảng 3.10: Bảng booking\_history*

# CHƯƠNG 4: KẾT QUẢ CÀI ĐẶT CHƯƠNG TRÌNH

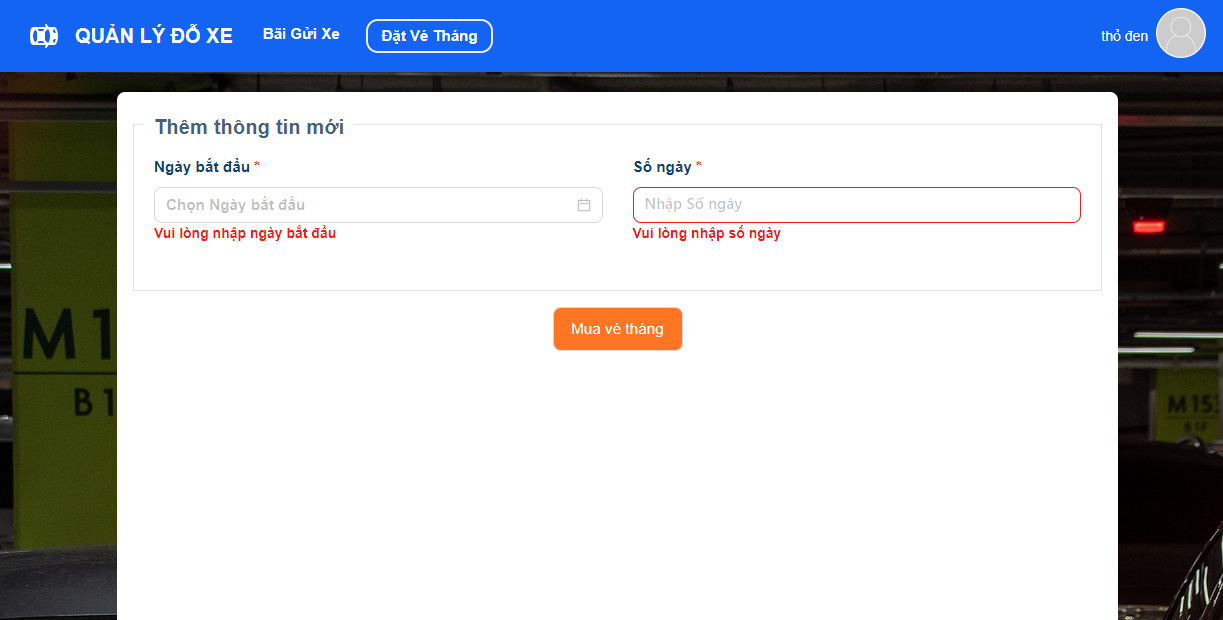
## 4.1. Chức năng dành cho người thuê bãi đỗ xe

4.1.1. Giao diện chính



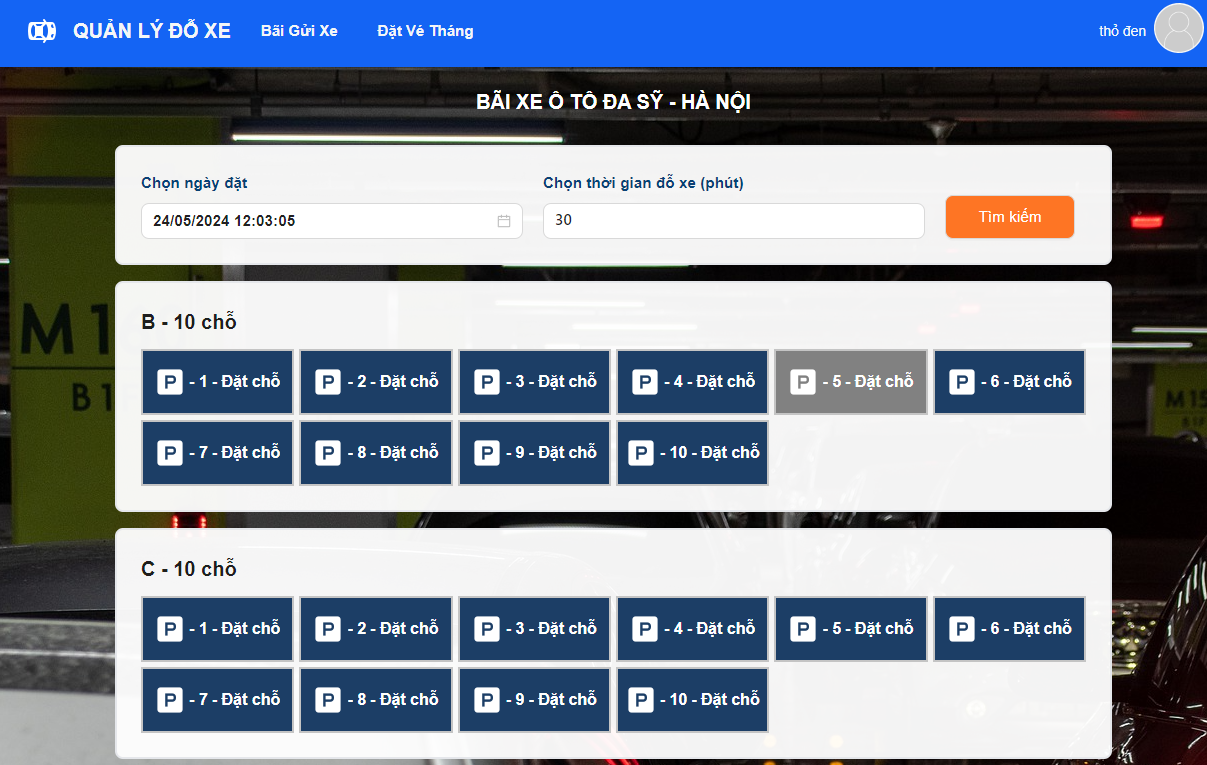
Hình 3.32: Hình ảnh giao diện chính

4.1.2. Giao diện đặt vé tháng



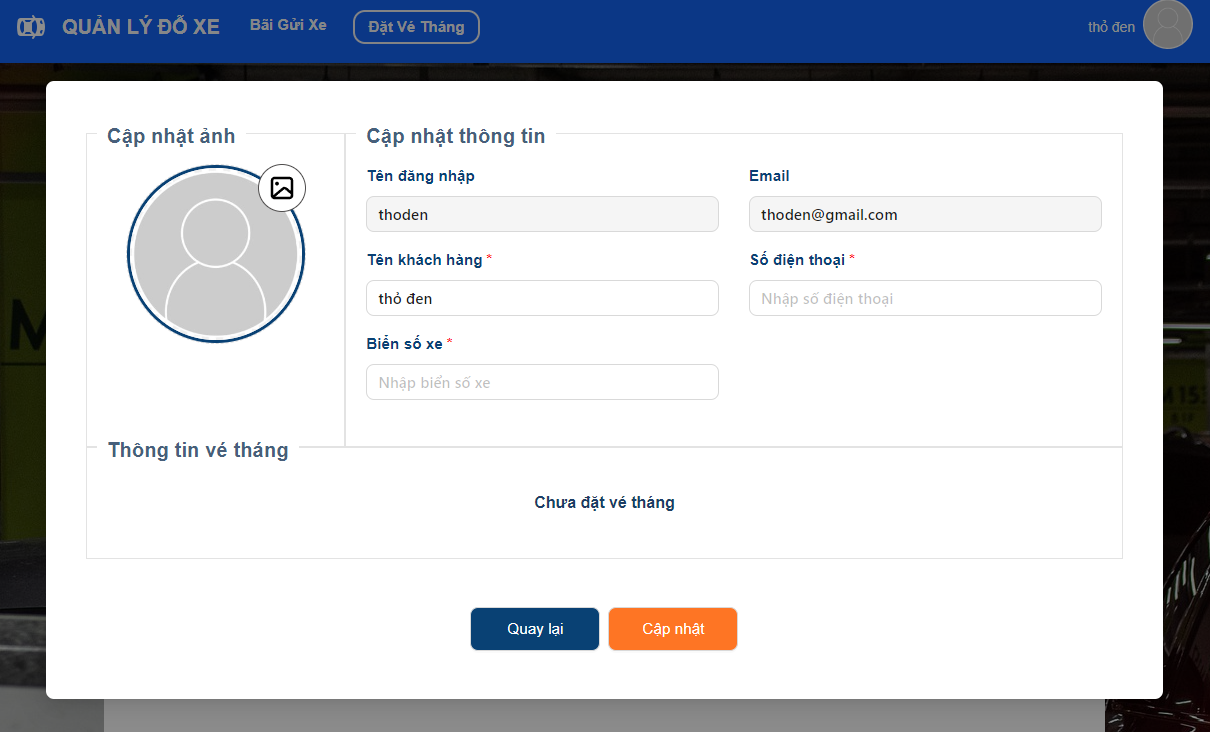
Hình 3.33: Hình ảnh giao diện đặt vé tháng

4.1.3. Giao diện đặt chỗ đỗ xe



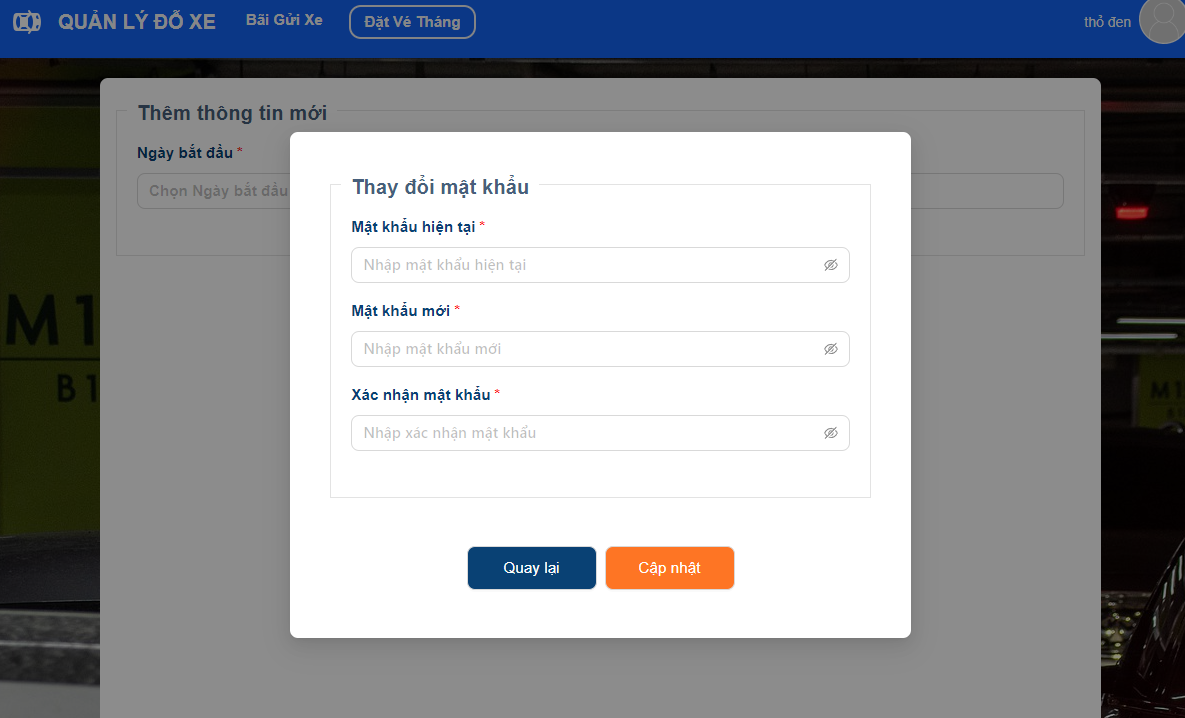
Hình 3.34: Hình ảnh giao diện đặt chỗ đỗ xe

4.1.4. Giao diện cập nhật thông tin tài khoản



Hình 3.35: Hình ảnh giao diện cập nhật thông tin tài khoản

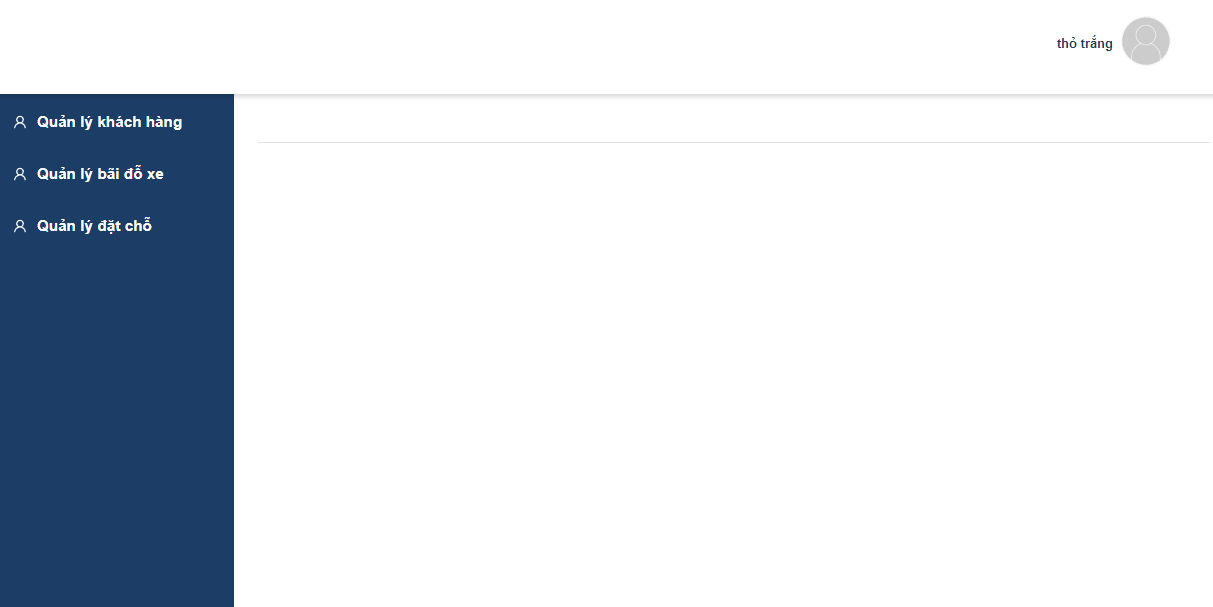
4.1.4. Giao diện đổi mật khẩu



Hình 3.36: Hình ảnh giao diện đổi mật khẩu

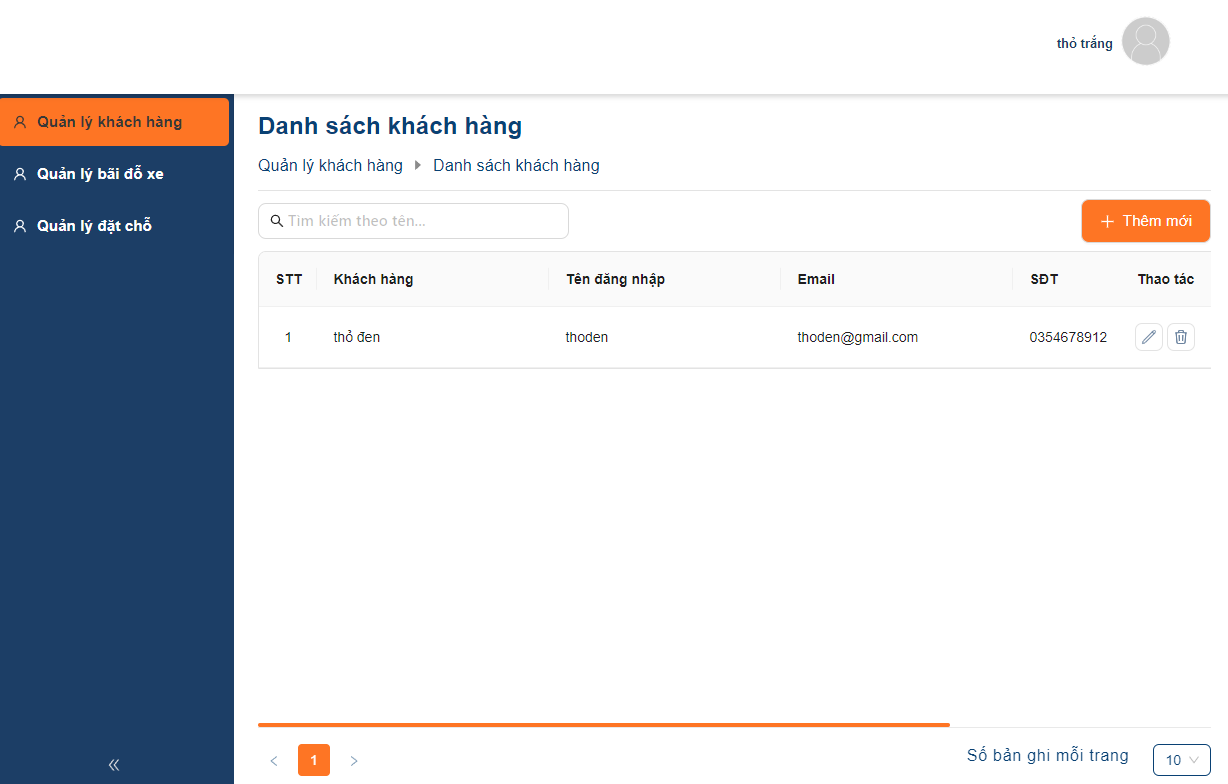
## 4.2. Chức năng dành cho admin

## 4.2.1. Giao diện chính



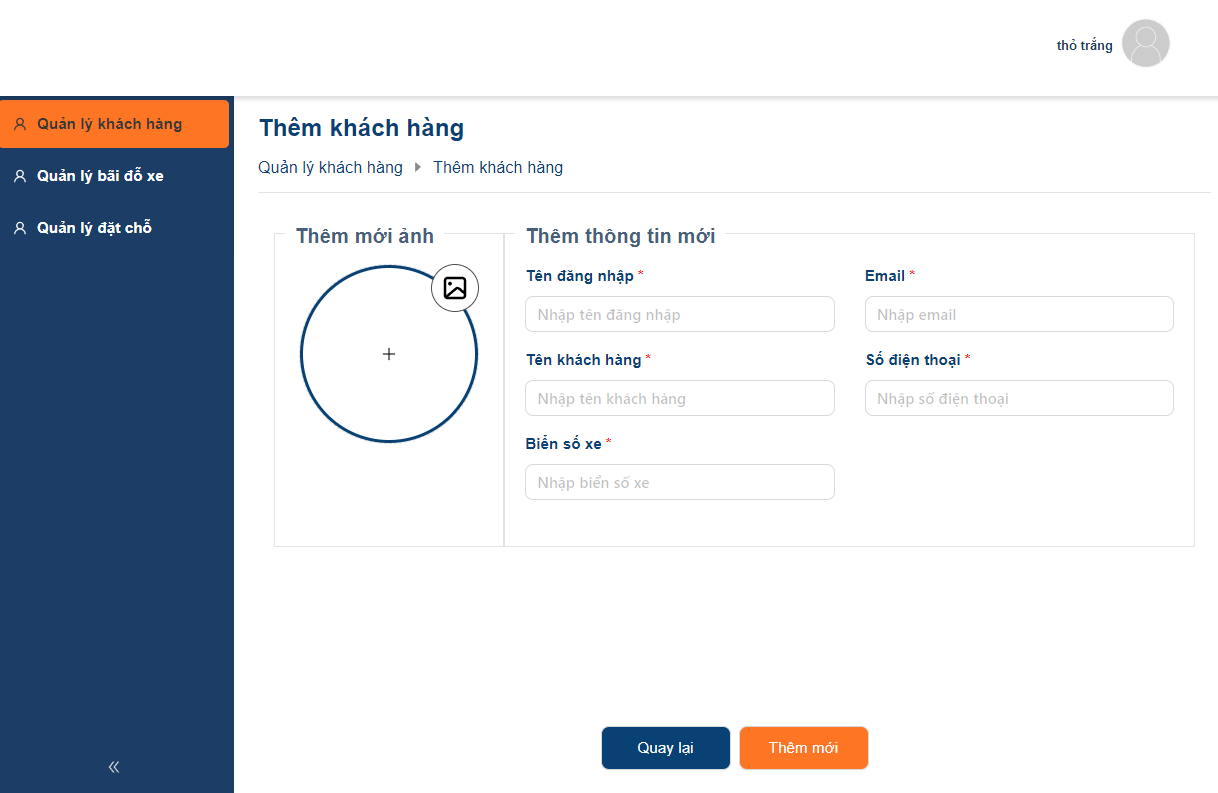
Hình 3.37: Hình ảnh giao diện chính của admin

## 4.2.1. Giao diện quản lý khách hàng



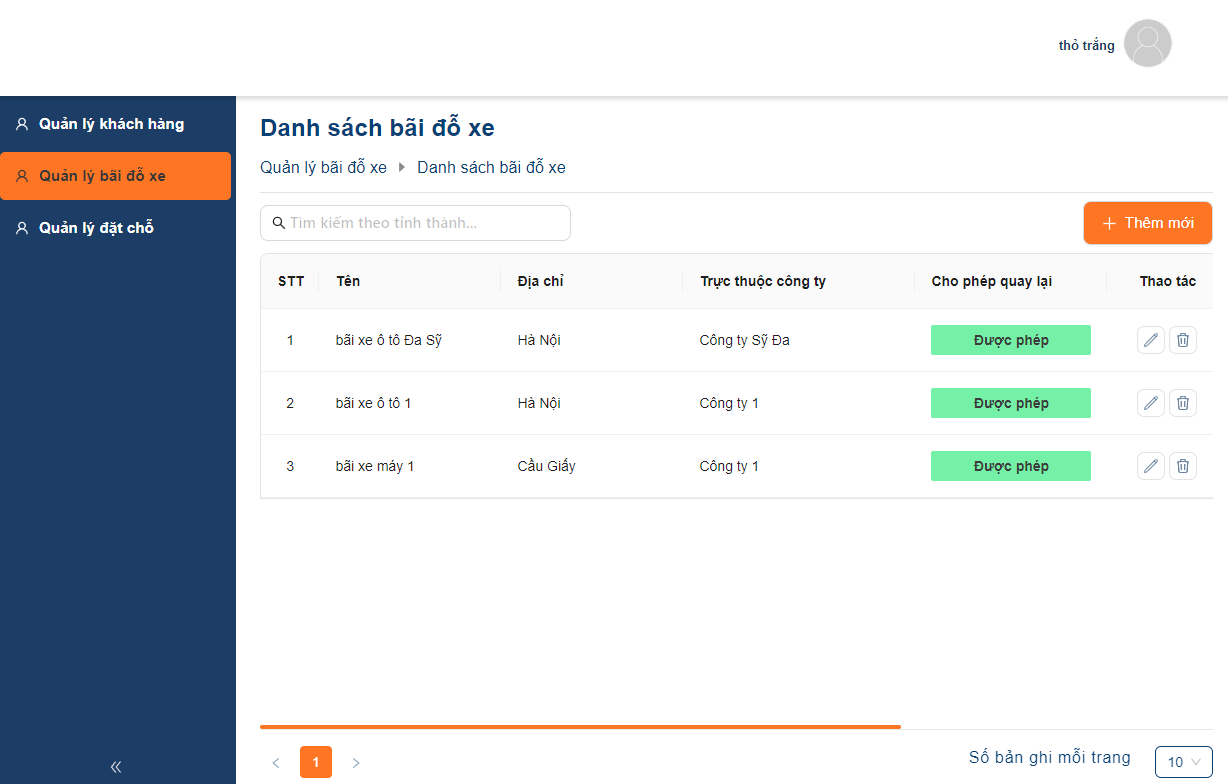
Hình 3.38: Hình ảnh giao diện quản lý khách hàng

## 4.2.2. Giao diện thêm khách hàng

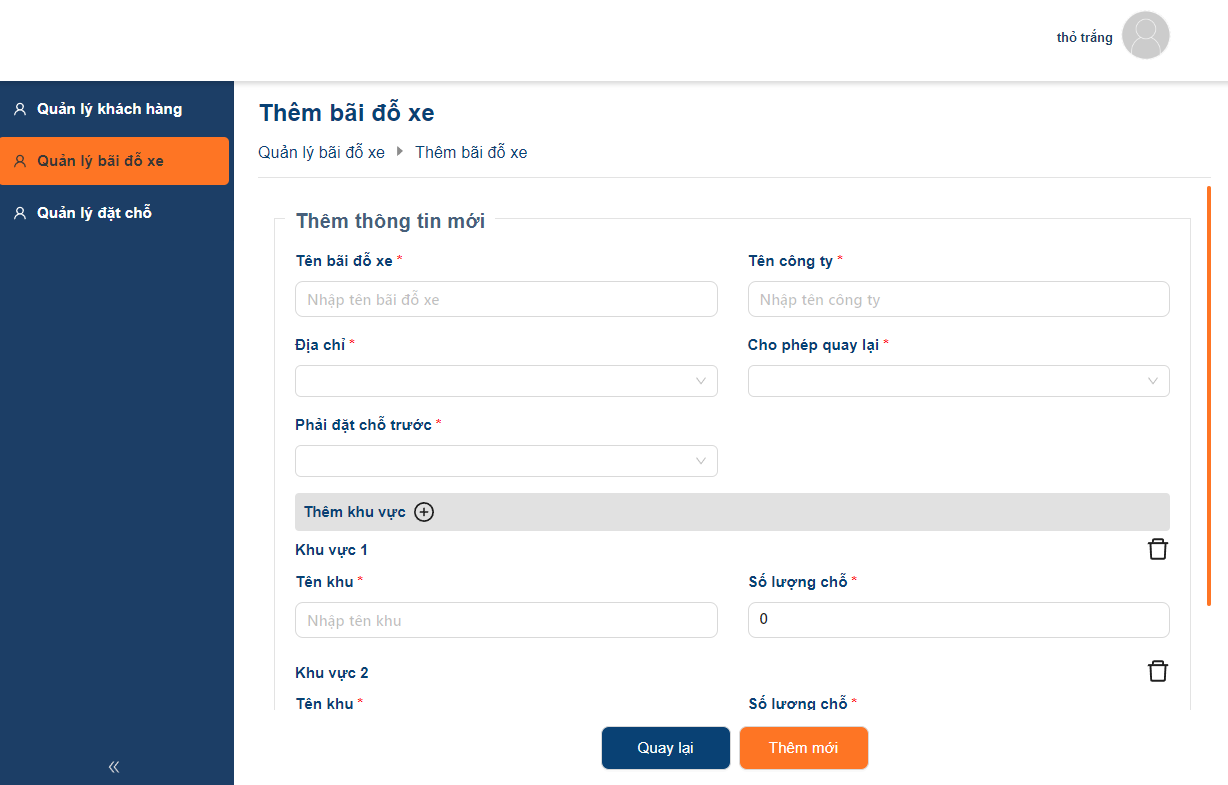


Hình 3.39: Hình ảnh giao diện thêm khách hàng

## 4.2.2. Giao diện quản lý bãi đỗ

*Hình* 3.40: Hình ảnh giao diện quản lý bãi đỗ xe

## 4.2.2. Giao diện quản lý bãi đỗ



*3.41: Hình ảnh giao diện các thao tác chính của bãi đỗ xe*

**KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

Đề tài “**Nghiên cứu xây dựng hệ thống website quản lý bãi đỗ xe ô tô**” cũng xuất phát từ thực tế ngày nay nhằm tạo một nền tảng cơ sở ban đầu để có thể hỗ trợ thêm cho các bạn muốn thiết kế một Website cho công ty hay một Website cá nhân.

Mặc dù đã có nhiều cố gắng, tìm hiểu các kiến thức đã học, kết hợp tra cứu các tài liệu chuyên nghành nhưng do còn hạn chế về thời gian, khả năng và kinh nghiệm nên không tránh khỏi những thiếu sót nhất định nên đề tài đã hoàn thành ở mức độ sau:

* Tìm hiểu ngôn ngữ lập trình Web Java Spring Boot
* Tìm hiểu chi tiết về phân tích thiết kế hệ thống
* Tìm hiểu chi tiết về hệ quản trị cơ sở dữ liệu MySQL
* Áp dụng xây dựng ứng dụng thực nghiệm trang Website hệ thống quản lý bãi đỗ xe ô tô.
* Triển khai website nền web thành công, tương thích với đầy đủ trình duyệt như:Cốc cốc, chrome, edge,…

**Hướng nghiên cứu phát triển:**

* Tìm hiểu sâu hơn về ngôn ngữ Java và hệ quản trị cơ sở dữ liệu MySQL để có thể đáp ứng nhiều hơn nữa nhu cầu của người sử dụng, phát triển và tối ưu hóa hệ thống,
* Kết hợp ngôn ngữ Java với các hệ quản trị cơ sở dữ liệu lớn hơn: SQL Server, Oracle…
* Tìm hiểu thêm một số ngôn ngữ, các phần mềm ứng dụng để nâng cao giao diện đồ họa đẹp mắt, thân thiện hơn…
* Xây dựng trang Web quy mô lớn hơn với nhiều ứng dụng, chức năng...