BỘ CÔNG THƯƠNG

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**BÁO CÁO TỔNG KẾT**

**ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN**

**NGHIÊN CỨU PHÁT TRIỂN ỨNG DỤNG TÌM BÃI ĐỖ XE THÔNG MINH TRÊN NỀN HỆ ĐIỀU HÀNH ANDROID AUTOMOTIVE**

**Sinh viên thực hiện:**

**Thân Ngọc Thiện Lớp HTTT01, Khoa CNTT**

Nguyễn Hoàng Dũng Lớp HTTT01, Khoa CNTT

Nguyễn Thế Đạt Lớp HTTT01, Khoa CNTT

Đỗ Quang Lâm Lớp HTTT01, Khoa CNTT

Nguyễn Tiến Mạnh Lớp CNTT06, Khoa CNTT

**Người hướng dẫn: TS. Phạm Văn Hà**

**Hà Nội, 5/2023**

BỘ CÔNG THƯƠNG

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**BÁO CÁO TỔNG KẾT**

**ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN**

**NGHIÊN CỨU PHÁT TRIỂN ỨNG DỤNG TÌM BÃI ĐỖ XE THÔNG MINH TRÊN NỀN HỆ ĐIỀU HÀNH ANDROID AUTOMOTIVE**

**Sinh viên thực hiện:**

**Thân Ngọc Thiện. Giới tính: Nam. Dân tộc: Kinh**

**Lớp, khoa: Hệ thống thôn tin 1, Công nghệ thông tin Năm thứ: 2/4**

**Ngành học: Hệ thống thông tin**

Nguyễn Hoàng Dũng. Giới tính: Nam. Dân tộc: Kinh

Lớp, khoa: Hệ thống thông tin 1, Công nghệ thông tinNăm thứ: 2/4

Ngành học: Hệ thống thông tin

Nguyễn Thế Đạt. Giới tính: Nam. Dân tộc: Kinh

Lớp, khoa: Hệ thống thông tin 1, Công nghệ thông tin Năm thứ: 2/4

Ngành học: Công nghệ thông tin

Đỗ Quang Lâm. Giới tính: Nam. Dân tộc: Kinh

Lớp, khoa: Hệ thống thông tin 1, Công nghệ thông tin Năm thứ: 2/4

Ngành học: Công nghệ thông tin

Nguyễn Tiến Mạnh. Giới tính: Nam. Dân tộc: Kinh

Lớp, khoa: Công nghệ thông tin 6, Công nghệ thông tin Năm thứ: 2/4

Ngành học: Công nghệ thông tin

**Người hướng dẫn: TS. Phạm Văn Hà**

**Hà Nội, 5/2023**

# MỤC LỤC

[MỤC LỤC 1](#_Toc135650933)

[DANH MỤC HÌNH ẢNH 1](#_Toc135650934)

[MỞ ĐẦU 3](#_Toc135650935)

[Chương 1: TỔNG QUAN ANDROID AUTOMOTIVE VÀ VẤN ĐỀ TÌM BÃI ĐỖ XE 7](#_Toc135650936)

[1.1 Tổng quan về hệ điều hành Android Automotive 7](#_Toc135650937)

[1.1.1 Android Automotive là gì ? 7](#_Toc135650938)

[1.1.1.1 Android Automotive hỗ trợ những tiện ích nào ? 12](#_Toc135650939)

[1.1.1.2 Có bao nhiêu hãng xe đang sử dụng Android Automotive ? 14](#_Toc135650940)

[1.1.1.3 Android Automotive có gì khác so với Android Auto? Lựa chọn nào tốt hơn ? 16](#_Toc135650941)

[1.1.1.4 Android Automotive sẽ là tương lai của hệ thống giải trí trên xe sau này 18](#_Toc135650942)

[1.1.2 Tổng quan về vấn đề tìm bãi đỗ xe 20](#_Toc135650943)

[1.1.2.1 Thực trạng số lượng xe ô tô và số lượng bãi đỗ hiện nay 20](#_Toc135650944)

[1.1.2.2 Những hậu quả mà sự chênh lệch số xe và bãi đỗ để lại. 25](#_Toc135650945)

[Chương 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 29](#_Toc135650946)

[2.1. Xác định yêu cầu 29](#_Toc135650948)

[2.2. Đưa ra các chức năng của hệ thống 29](#_Toc135650949)

[2.3. Sơ đồ Use case 31](#_Toc135650950)

[2.4. Phân tích dặc tả Use case 32](#_Toc135650951)

[2.5. Thiết kế cơ sở dữ liệu 40](#_Toc135650952)

[2.6. Một số giao diện hệ thống 44](#_Toc135650953)

[Chương 3: KẾT QUẢ NGHIÊN CỨU 46](#_Toc135650954)

[3.1.1 Các công cụ, công nghệ sử dụng để xây dựng ứng dụng 46](#_Toc135650955)

[3.1.2 Chương trình và kết quả 53](#_Toc135650956)

[KẾT LUẬN 59](#_Toc135650957)

[TÀI LIỆU THAM KHẢO 62](#_Toc135650958)

# DANH MỤC HÌNH ẢNH

[Hình 1‑1: Kiến trúc lớp trừu tượng của Android Automotive 12](#_Toc135650915)

[Hình 2‑1: Sơ đồ Use case phía backend 31](#_Toc135650916)

[Hình 2‑2: Sơ đồ Use case phía FrontEnd 32](#_Toc135650917)

[Hình 2‑3: Sơ đồ cơ sở dữ liệu của hệ thống 40](#_Toc135650918)

[Hình 2‑4: Trang đăng ký của hệ thống 44](#_Toc135650919)

[Hình 2‑5: Trang đăng nhập của hệ thống 44](#_Toc135650920)

[Hình 2‑6: Màn hình chính của khách hàng khi sử dụng ứng dụng 45](#_Toc135650921)

[Hình 3‑1 Ứng dụng Google Map với thuật toán Dijkstra 52](#_Toc135650922)

[Hình 3‑2: Cài đặt cấu hình bảo vệ đăng nhập 53](#_Toc135650923)

[Hình 3‑3: Cài đặt xử lý"Admin" trong hệ thống 55](#_Toc135650924)

[Hình 3‑4: Cài đặt xử lí User trong hệ thống 56](#_Toc135650925)

[Hình 3‑5:Cài đặt xử lí yêu cầu liên quan đến bãi đỗ xe 57](#_Toc135650926)

# MỞ ĐẦU

1. Đề tài và lý do lựa chọn đề tài

Với sự phát triển nhanh chóng của công nghệ ô tô và sự gia tăng nhu cầu về giao thông thông minh, việc tìm kiếm và quản lý bãi đỗ xe đã trở thành một thách thức đối với người dùng. Với mục tiêu tạo ra một giải pháp tiện lợi và thông minh cho vấn đề này, nhóm nghiên cứu đã chọn phát triển ứng dụng đỗ xe thông minh trên nền tảng hệ điều hành Android Automotive. Điều này không chỉ mang lại lợi ích cho người dùng và xã hội mà còn tạo cơ hội khám phá những lĩnh vực mới của công nghệ ô tô và giao thông thông minh.

Vấn đề đỗ xe đã trở thành một thách thức đối với các thành phố lớn trên toàn thế giới. Bằng cách phát triển một ứng dụng tìm bãi đỗ xe thông minh, có thể giúp giảm tắc nghẽn giao thông, tiết kiệm thời gian và năng lượng, đồng thời giảm thiểu ô nhiễm môi trường. Vấn đề đỗ xe đã trở thành một thách thức đối với các thành phố lớn trên toàn thế giới. Bằng cách phát triển ứng dụng tìm chỗ đỗ xe thông minh, bạn có thể giúp giảm tắc nghẽn giao thông, tiết kiệm thời gian và năng lượng cũng như giảm ô nhiễm môi trường.

Giải pháp bãi giữ xe thông minh là một giải pháp rất phù hợp và cần thiết cho xã hội ngày nay. Việc tạo một ứng dụng dựa trên Android Automotive có thể giúp thu hút sự chú ý của người dùng và các tổ chức tương ứng, đồng thời cung cấp một cách liền mạch để xác định vị trí và đặt chỗ đỗ xe.

Như vậy, phát triển ứng dụng tìm kiếm bãi đậu xe thông minh cho Android Automotive có thể mang lại lợi ích cho người dùng cũng như xã hội. Điều này cũng tạo cơ hội để nghiên cứu và phát triển hơn nữa trong lĩnh vực công nghệ ô tô cũng như hệ thống giao thông thông minh.

1. Mục tiêu của đề tài

Bằng cách khai thác tiềm năng của Android Automotive OS, việc phát triển ứng dụng định vị bãi đỗ xe thông minh trở thành một lựa chọn hấp dẫn và có tiềm năng đáng kể. Sự kết hợp giữa công nghệ ô tô và hệ điều hành Android do Google phát triển cho phép nhóm nghiên cứu nắm bắt và tận dụng những lợi thế thiết yếu của nền tảng này để tạo ra một giải pháp sáng tạo và thuận tiện cho việc tìm kiếm và quản lý chỗ đỗ xe. Tính linh hoạt và khả năng tích hợp sâu của Android Automotive vào các hệ thống ô tô hiện đại mang đến nhiều cơ hội để sáng tạo và tiến bộ hơn nữa trong việc mang lại trải nghiệm tốt nhất cho người dùng, giúp giải quyết một trong những vấn đề giao thông phổ biến và phổ biến nhất trên thế giới. Bằng cách khai thác sức mạnh của hệ điều hành Android Automotive, nhóm có thể biến thành một ứng dụng tìm chỗ đỗ xe thông minh, đáng tin cậy và hiệu quả giúp hợp lý hóa việc đi lại và cải thiện chất lượng cuộc sống thành phố hiện đại. bãi đậu xe.

Mục tiêu chính đầu tiên của nghiên cứu này là hiểu rõ về hệ điều hành Android Automotive và khám phá tiềm năng của nó trong việc phát triển ứng dụng tìm bãi đỗ xe thông minh. Qua việc nghiên cứu sâu về hệ điều hành này, nhóm mong muốn khám phá các khả năng tích hợp, giao tiếp và tương tác của Android Automotive. Mục tiêu thứ hai tạo ra một ứng dụng đỗ xe thông minh tận dụng sức mạnh của Hệ điều hành Android Automotive. Mục tiêu thứ ba là phát triển một ứng dụng định vị chỗ đỗ xe dựa trên Android Automotive OS sử dụng API Google Maps và được đồng bộ hóa với điều hướng của điện thoại, dịch vụ định vị và các ứng dụng khác. Mục tiêu cuối cùng là phát triển các công cụ để tạo ra một ứng dụng có thể được phát triển bởi B2B, B2C hoặc tất cả các mô hình kinh doanh cho nhu cầu riêng của họ, bao gồm tất cả các khía cạnh trong phạm vi của nghiên cứu này trong một gói gắn kết.

Việc nghiên cứu và phát triển các ứng dụng trên nền tảng Android Automotive vẫn đang ở giai đoạn đầu, điều này mang lại một môi trường vô cùng rộng lớn cho tư duy sáng tạo và hợp tác cũng như áp dụng các công nghệ mới liên quan đến ngành công nghiệp ô tô và giao thông thông minh. Việc xây dựng ứng dụng trên nền tảng này giúp giải quyết các vấn đề liên quan đến đỗ xe dễ dàng hơn, chẳng hạn như tìm chỗ, nhận thông tin về giá và khả năng tiếp cận cũng như tích hợp thanh toán kỹ thuật số. Nó giúp cải thiện trải nghiệm người dùng và tối ưu hóa toàn bộ quá trình từ đầu đến cuối.

1. Nội dung nghiên cứu

* Tìm hiểu về hệ điều hành Android Automotive: Tìm hiểu tổng quan về hệ điều hành Android Automotive như: Android Automotive OS là gì? Hỗ trợ những tiện ích nào, so sánh với Android Auto, …
* Các vấn đề xoay quanh bãi đỗ xe hiện nay: Thực trạng, những ảnh hưởng, tác động của việc đặt xe đến những vấn đề xung quanh.
* Xây dựng thiết kế hệ thống tìm bãi đỗ xe.
* Cài đặt và kiểm thử hệ thống.

1. Phạm vi đề tài

Trọng tâm của nghiên cứu này là nghiên cứu chi tiết về hệ điều hành Android Automotive bao gồm: cấu trúc, giao diện lập trình ứng dụng và các tính năng chính. Tiếp đến là phát triển ứng dụng tìm chỗ đỗ xe thông minh trên nền tảng hệ điều hành Android Automotive. Các yêu cầu và tính năng của ứng dụng đỗ xe thông minh sau đó được xác định thông qua phỏng vấn và khảo sát người dùng.

Tiếp theo, nghiên cứu sẽ tập trung vào thiết kế giao diện và trải nghiệm người dùng để tạo ra một giao diện thân thiện và dễ sử dụng cho việc tìm kiếm và quản lý bãi đỗ xe thông minh.

Sau đó, nghiên cứu sẽ tiến hành phát triển và triển khai ứng dụng tìm kiếm chỗ đỗ xe thông minh trên nền tảng Android Automotive. Các chức năng tìm kiếm, đánh giá và thanh toán được lập trình và tích hợp vào hệ thống xe thông minh.

Cuối cùng, là một phần của nghiên cứu, các thử nghiệm được chạy để đánh giá ứng dụng và kiểm tra tính ổn định, hiệu suất và chức năng của nó. Thu thập phản hồi từ người dùng thực để đo lường hiệu quả và giúp chúng tôi cải thiện trong tương lai.

1. Phương pháp nghiên cứu

Trong việc nghiên cứu và phát triển ứng dụng tìm bãi đỗ xe thông minh trên nền hệ điều hành Android Automotive một số phương pháp nghiên cứu được áp dụng là:

* Phương pháp nghiên cứu tài liệu.
* Phương pháp thực nghiệm.
* Phương pháp phân tích và tổng hợp.
* Phương pháp phân loại và hệ thống.

1. Bố cục báo cáo

Quyển báo cáo nghiên cứu khoa học sẽ bao gồm các chương:

Chương 1: Tổng quan về Android Automotive OS và các vấn đề về tìm bãi đỗ xe.

Chương 2: Phân tích và thiết kê hệ thống

Chương 3: Phương pháp thực hiện và kết quả

Phần kết luận

NỘI DUNG

# TỔNG QUAN ANDROID AUTOMOTIVE VÀ VẤN ĐỀ TÌM BÃI ĐỖ XE

## Tổng quan về hệ điều hành Android Automotive

### **Android Automotive là gì ?**

Android được ra mắt vào cuối năm 2008 như một hệ điều hành di động của Google [1]. Ban đầu, nền tảng mã nguồn mở này được thiết kế cho các điện thoại di động trang bị màn hình cảm ứng, hay còn gọi là điện thoại thông minh. Tuy nhiên, thành công của hệ điều hành đa năng này trên các điện thoại phổ biến đã dẫn đến việc Google phát triển các phiên bản Android cho TV và đồng hồ thông minh vào đầu những năm 2010. Việc thâm nhập của Android vào các thị trường khác nhau đã đưa đến việc ra mắt Android Auto vào năm 2015. Điều này cũng một phần là do sự ra đời của màn hình cảm ứng cho hệ thống thông tin giải trí trên xe (IVI). Android Auto là một ứng dụng chạy trên điện thoại di động và khi được kết nối với IVI (qua USB, WiFi hoặc Bluetooth AVRCP) chiếu một số ứng dụng đã chọn lên màn hình IVI. Trọng tâm của Android Auto là cung cấp các ứng dụng đa phương tiện và điều hướng với trải nghiệm người dùng cải tiến để giảm sự phân tâm của tài xế.

Mặc dù được hỗ trợ bởi tất cả các OEM chính (và Apple Carplay) cho hầu hết các mẫu xe của họ từ năm 2019, điểm hạn chế lớn nhất của Android Auto là thiếu quyền truy cập vào bất kỳ dữ liệu nào được tạo ra bên trong xe. Nó chỉ dựa vào các cảm biến trên thiết bị cầm tay và không đọc hoặc viết bất kỳ dữ liệu nào vào mạng trong xe (IVN). Do đó, tích hợp đầy đủ của ô tô với Android Auto không thể thực hiện được. Mô hình kinh doanh của Android là một phần mở rộng của mô hình kinh doanh Google hiện tại: Doanh thu được thu được từ Tìm kiếm, AdSense và các ứng dụng liên quan đến nó là những công cụ giúp tạo ra thu nhập từ việc đăng quảng cáo trên website hoặc ứng dụng. Đối với hệ điều hành Android, Google cũng kiếm được tiền từ doanh thu bán ứng dụng trên Google Play Store, phí cấp phép (Google Play Services) cũng như nội dung đa phương tiện trên Google Play như âm nhạc.

Tuy nhiên, đối với trường hợp sử dụng trong ô tô, điều này không hoàn toàn khả thi với Android Auto vì không thể tận dụng được nhiều dữ liệu hơn. Đây là lý do tại sao Google giới thiệu Android Automotive tại sự kiện Google I/O 2018. Họ đã thông báo về việc hợp tác với liên minh sản xuất ô tô lớn của Renault-Nissan-Mitsubishi để cài đặt hệ thống giải trí thông tin được cung cấp bởi Android Automotive trên hàng triệu chiếc xe bắt đầu từ năm 2021[2]. Mặc dù các nhà sản xuất ô tô đã ngại chia sẻ dữ liệu quý giá về xe hơi với Google, nhưng nỗ lực của Google trong việc tạo ra một hệ thống điều hành sạch sẽ, mạnh mẽ được thiết kế để chạy độc lập trên hệ thống giải trí thông tin đã thuyết phục các nhà sản xuất ô tô đưa công nghệ mới này vào sử dụng. Một cửa hàng Google Play cụ thể cho các loại xe sẽ cho phép triển khai các ứng dụng bên thứ ba trên nhiều loại xe khác nhau độc lập với nhà sản xuất và có thể cho phép các nhà sản xuất ô tô dễ dàng truy cập vào một phần lợi nhuận với Google. Các ứng dụng bên thứ ba yêu cầu truy cập vào hệ thống mạng thông tin nội bộ của xe (IVN) có thể bao gồm các ứng dụng nhà thông minh để tối ưu hóa quản lý sạc pin trong ga-ra nhà của khách hàng hoặc các ứng dụng bảo hiểm dựa trên việc sử dụng (UBI). Các ứng dụng này tính toán hành vi lái xe từ một tập hợp các cảm biến, chẳng hạn như tốc độ hoặc phanh, để tự động điều chỉnh phí bảo hiểm. Vậy Android Automotive được Google giới thiệu như thế nào?

Android Automotive là nền tảng Android cơ bản chạy các ứng dụng Android hệ thống IVI được cài đặt sẵn cũng như các Ứng dụng Android tùy chọn của bên thứ hai và thứ ba.

Android Automotive cung cấp tính mở, khả năng tùy chỉnh và mở rộng quy mô cho các hệ thống thông tin giải trí trên ô tô và các thiết bị đầu. Tính cởi mở mang lại hiệu quả mới bằng cách cung cấp các tính năng thông tin giải trí cơ bản trên ô tô trong một cơ sở mã nguồn mở và miễn phí. Tùy chỉnh cho phép người triển khai phân biệt sản phẩm khi họ thấy phù hợp. Quy mô đạt được thông qua khung, ngôn ngữ và API chung của Android, tất cả đều cho phép sử dụng lại kiến ​​thức chuyên môn phát triển và phần mềm đã hoàn thiện từ hàng trăm nghìn nhà phát triển Android trên toàn thế giới.

 Android Automotive lần đầu được giới thiệu vào tháng 3/2017 do Google, Intel, Volvo, Audio cùng phát triển.Trong những năm gần đây, kết nối không dây trên ô tô đã trở nên phổ biến. Các chức năng hiện có của các phương tiện kết nối (CV) bao gồm thông tin giải trí, an toàn, hiệu suất chẩn đoán, điều hướng và thanh toán. Trong giai đoạn tiếp theo của CV, các xe sẽ kết nối với các dịch vụ bên thứ ba bằng cách sử dụng kết nối dữ liệu tích hợp sẵn, giới thiệu các nền tảng thu thập dữ liệu phương tiện mới, chẳng hạn như BMW Car Data. Ước tính hiện có khoảng 200 triệu ô tô được kết nối với web, và con số này được dự báo sẽ tăng lên gần 500 triệu chỉ riêng ở Mỹ, châu Âu và Trung Quốc vào 2025.

Theo Google, Android Automotive được tích hợp vào xe bởi các nhà sản xuất ô tô – tức là ta không thể mua một chiếc tablet Android, lắp nó vào xe và gọi đó là “Android Automotive” được. Vì chỉ mới được Google công bố vào năm 2017, hầu hết ô tô tích hợp giao diện Android Automotive đều thuộc đời 2021 trở về sau. Một số nhà sản xuất ô tô nổi tiếng bán xe kèm Android Automotive bao gồm Ford, GM, Volvo, Renault, Rivian, và Lucid Motors[4].

Tiếp theo, chúng tôi sẽ giới thiệu ngăn xếp Android Automotive trong mã nguồn của Google, trình bày về mô hình quyền truy cập hiện có cho các phương tiện, cung cấp một hướng dẫn về giao thức CAN, cũng như thảo luận về VHAL - thành phần quan trọng nhất trong việc giao tiếp với xe hơi. Android Automotive Stack là một phần mềm đa lớp, được xây dựng trên nền tảng Android, được Google phát triển đặc biệt cho các phương tiện thông minh. Ngăn xếp này cung cấp các tính năng và API được tùy chỉnh cho các ứng dụng và dịch vụ phổ biến nhất trong các phương tiện, như điều khiển âm thanh, điều hòa không khí và hệ thống định vị.

Mô hình quyền truy cập cho các phương tiện trong Android Automotive được thiết kế dựa trên mô hình quyền truy cập cho các ứng dụng di động. Các ứng dụng được phân loại thành các loại quyền khác nhau, ví dụ như quyền truy cập vào tệp hoặc quyền truy cập vào vị trí của người dùng. Mỗi loại quyền được xác định bởi một nhóm các quyền con cụ thể.

Giao thức CAN (Controller Area Network) là giao thức truyền thông phổ biến nhất trong các hệ thống điện tử trong xe hơi. Giao thức này được thiết kế để cho phép các thành phần điện tử trong xe hơi như động cơ, hệ thống giảm xóc và hệ thống phanh giao tiếp với nhau. Dữ liệu được truyền tải qua các bus CAN, với mỗi bus có thể có nhiều thành phần điện tử được kết nối.

VHAL (Vehicle Hardware Abstraction Layer) là thành phần quan trọng nhất trong việc giao tiếp giữa các ứng dụng Android và các phần cứng của phương tiện. VHAL cung cấp các API để truy cập các chức năng phần cứng của xe hơi như cảm biến và đèn xe. VHAL được thiết kế để ẩn giấu chi tiết phần cứng cụ thể của từng phương tiện, giúp đơn giản hóa việc phát triển các ứng dụng và dịch vụ trên các nền tảng khác nhau của các nhà sản xuất phương tiện.

Tổng quan về Android Automotive Stack, mô hình quyền truy cập, giao thức CAN và VHAL cung cấp cho chúng ta một cái nhìn tổng quan về cách Android Automotive hoạt động và cách các ứng dụng và dịch vụ được phát triển trên nền tảng này có thể tương tác với các phần cứng của phương tiện. Tuy nhiên, để đảm bảo an toàn và bảo mật của hệ thống, cần phải thực hiện một kiểm tra

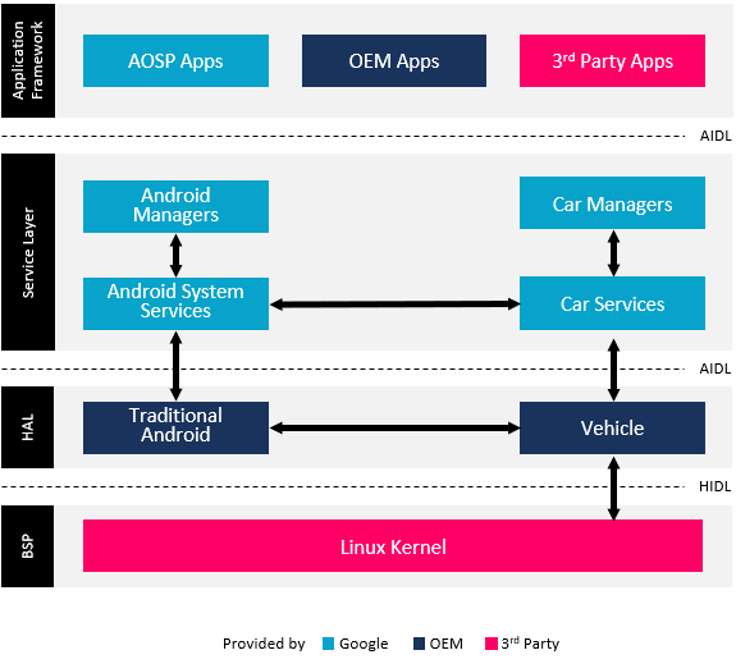
Kiến trúc của hệ điều hành Android Automotive được chia làm bốn lớp, bao gồm các thành phần chung như sau[5]:

Application Framework: Đây còn được gọi là lớp HMI (Human Machine Interface) và nó chứa các ứng dụng hệ thống và người dùng. Lý tưởng nhất là thiết kế các ứng dụng theo cách sao cho hầu hết các yêu cầu kinh doanh cốt lõi được chuyển sang lớp Services. Thiết kế như vậy tạo điều kiện cho khả năng mở rộng trong tương lai và cập nhật dễ dàng.

Android Automotive System Services: Là một tập hợp các dịch vụ và API (Application Programming Interface) được cung cấp bởi hệ điều hành Android của Google được thiết kế đặc biệt để sử dụng trong các ứng dụng ô tô. Điều này bao gồm các dịch vụ liên quan đến quản lý dữ liệu phương tiện, chẳng hạn như tốc độ, vị trí và mức nhiên liệu của phương tiện, cũng như các dịch vụ quản lý phát lại âm thanh và các chức năng phương tiện khác.

Hardware Abstraction Layer (HAL): HAL hiển thị các giao diện ô tô cho các dịch vụ hệ thống theo cách đạt được kiến trúc bất khả tri về phương tiện. Khung ứng dụng, Dịch vụ hệ thống và HAL là các thành phần cốt lõi của nền tảng Hệ điều hành Android dành cho ô tô và các lớp này hỗ trợ trao đổi dữ liệu giữa các ứng dụng và ECUs của xe.

Linux Kernel: Linux là nhân cơ bản của kiến trúc Android Automotive.



Hình 1‑1: Kiến trúc lớp trừu tượng của Android Automotive

#### Android Automotive hỗ trợ những tiện ích nào ?

Android Automotive là một hệ thống giải trí thông tin và điều khiển thông minh được tích hợp trên các chiếc xe hơi. Nó cung cấp cho người dùng nhiều tiện ích và tính năng, bao gồm[3]:

Android Automotive có khả năng điều khiển các chức năng của xe như điều hòa, bật đèn, khóa cửa.... Thông qua các nút bấm trên màn hình hoặc điều khiển giọng nói. Việc này giúp người dùng có thể dễ dàng điều khiển các chức năng của xe một cách tiện lợi và nhanh chóng, giúp tăng cường trải nghiệm lái xe và giảm thiểu nguy cơ tai nạn giao thông. Tuy nhiên, để sử dụng được các tính năng này, xe cần phải được trang bị các cảm biến và bộ điều khiển tương ứng, và các tính năng này phụ thuộc vào từng dòng xe cụ thể.

Truy cập vào các ứng dụng phổ biến: Android Automotive cung cấp cho người dùng khả năng truy cập vào các ứng dụng phổ biến như Google Maps, Spotify, Google Assistant và nhiều ứng dụng khác thông qua màn hình cảm ứng trên xe hơi. Nhờ tính năng này, người dùng có thể tìm kiếm địa điểm, lắng nghe nhạc, tìm kiếm thông tin và thực hiện các tác vụ khác trong khi lái xe.

Hỗ trợ điều khiển giọng nói: Android Automotive hỗ trợ điều khiển giọng nói, cho phép người dùng điều khiển hệ thống giải trí thông qua lệnh giọng nói. Tính năng này giúp người dùng tập trung lái xe mà vẫn có thể sử dụng các tính năng của hệ thống, giúp tăng cường an toàn khi lái xe.

Kết nối di động: Hệ thống Android Automotive cho phép người dùng kết nối với điện thoại thông minh của mình để truy cập vào danh bạ, cuộc gọi, tin nhắn và nhiều tính năng khác. Người dùng có thể sử dụng tính năng này để thực hiện cuộc gọi, gửi tin nhắn và kiểm tra thông tin mà không cần phải dùng tay.

Tính năng an toàn: Android Automotive cung cấp các tính năng an toàn để giúp người dùng lái xe an toàn hơn. Ví dụ, tính năng hỗ trợ đỗ xe sẽ cảnh báo người lái khi xe tiếp cận với chướng ngại vật, giúp người dùng đỗ xe an toàn hơn. Tính năng cảnh báo lùi xe sẽ thông báo cho người lái xe về các vật cản phía sau, giúp tránh va chạm và nguy hiểm khi lùi xe. Tính năng cảnh báo điểm mù sẽ cảnh báo người lái xe về các vật cản ở phía sau xe, giúp giảm nguy cơ tai nạn.

Tích hợp camera: Hệ thống Android Automotive có thể tích hợp các camera để giúp người dùng quan sát xung quanh xe và cung cấp hình ảnh chất lượng cao khi lái xe. Với tính năng này, người dùng có thể giám sát môi trường xung quanh, quan sát phía sau xe, giúp tránh va chạm và giảm nguy cơ tai nạn khi di chuyển.

Điều khiển từ xa: Hệ thống Android Automotive có tính năng điều khiển từ xa, cho phép người dùng kiểm soát một số tính năng của xe từ xa thông qua điện thoại di động. Ví dụ, người dùng có thể khóa hoặc mở cửa xe, bật hoặc tắt đèn xe và nhiều tính năng khác mà không cần phải đến gần xe.

Tính năng thông minh: Android Automotive có tính năng thông minh giúp tối ưu hóa trải nghiệm lái xe của người dùng. Ví dụ, tính năng này có thể đề xuất lộ trình đi lại tối ưu nhất cho người dùng, bao gồm thời gian và khoảng cách. Tính năng này cũng có thể giúp tối ưu hóa động cơ và tiết kiệm nhiên liệu.

Kết nối mạng: Hệ thống Android Automotive có khả năng kết nối mạng Wi-Fi, 4G hoặc 5G để người dùng có thể truy cập vào các dịch vụ trực tuyến, như truyền phát âm nhạc trực tuyến, cập nhật địa điểm và lộ trình, và nhiều tính năng khác. Kết nối mạng cũng cho phép hệ thống cập nhật phần mềm và bảo mật để đảm bảo an toàn khi sử dụng.

Tính năng kết nối với thiết bị gia đình: Hệ thống Android Automotive có tính năng kết nối với các thiết bị gia đình thông minh như Google Home và Nest. Tính năng này giúp người dùng có thể điều khiển các thiết bị gia đình từ xa, chẳng hạn như bật đèn, điều chỉnh nhiệt độ, và nhiều tính năng khác.

Hỗ trợ đa ngôn ngữ: Hệ thống Android Automotive hỗ trợ đa ngôn ngữ, cho phép người dùng chọn ngôn ngữ phù hợp với mình. Điều này giúp tăng cường trải nghiệm người dùng và giúp người dùng dễ dàng sử dụng các tính năng của hệ thống.

#### Có bao nhiêu hãng xe đang sử dụng Android Automotive ?

Xe hơi ngày nay không chỉ là phương tiện di chuyển mà còn được xem như một phần của cuộc sống hiện đại. Các nhà sản xuất xe hơi đang cố gắng tạo ra những chiếc xe thông minh hơn để đáp ứng nhu cầu của người dùng. Trong đó, hệ điều hành Android Automotive được xem là một trong những giải pháp tốt nhất để mang đến các tính năng thông minh và tiện ích trên các mẫu xe hơi.

-   **Tháng 9/2018**:[*Liên minh Renault-Nissan-Mitsubishi*](https://news.oto-hui.com/?s=Li%C3%AAn+minh+Renault-Nissan-Mitsubishi) *thông báo việc hợp tác với Google để đưa Android Automotive lên xe của họ từ năm 2021. Đây là một bước đột phá lớn trong lĩnh vực công nghệ ô tô và đã mở đường cho sự phát triển của Android Automotive trên các mẫu xe hơi.*

-   **Tháng 2/2019:** [*Polestar*](https://news.oto-hui.com/?s=Polestar) *(thương hiệu xe điện của Volvo) thương hiệu xe điện của Volvo, đã ra mắt chiếc xe điện Polestar 2. Điều đặc biệt là đây là chiếc xe đầu tiên tích hợp hệ điều hành Android Automotive và bắt đầu được bán ra vào năm 2020. Sự xuất hiện của Polestar 2 đã chứng minh rằng Android Automotive có thể được tích hợp trên các mẫu xe điện và đạt được thành công trong lĩnh vực này.*

-   **Tháng 4/2019**: *Google bắt đầu mở các API ra để nhà phát triển có thể bắt đầu xây dựng app cho Android Automotive. Điều này giúp tăng tính linh hoạt cho hệ điều hành này và mở ra nhiều cơ hội cho các ứng dụng và dịch vụ được tích hợp trên các mẫu xe hơi.*

-   **Tháng 9/2019**: *Đến lượt General Motors (tập đoàn ô tô nổi tiếng tại Mĩ) – nắm giữ nhiều thương hiệu nổi tiếng như Chevrolet, Cadillac …đã thông báo kế hoạch tương tự như liên minh Renault-Nissan-Mitsubishi để tích hợp hệ điều hành Android Automotive lên các mẫu xe của họ.*

-   **Tháng 2/2021**: *Ford thông báo sẽ dùng Android Automotive cho các dòng xe Ford và Lincoln từ năm 2023 trở đi. Điều này giúp cho các dòng xe của Ford và Lincoln được trang bị các tính năng thông minh và tiện ích thông qua kết nối với các ứng dụng trên điện thoại thông minh. Hệ điều hành này sẽ được tích hợp trên các mẫu xe mới của Ford và Lincoln từ năm 2023 trở đi, và sẽ được trang bị trên các mẫu xe của hãng ở thị trường Bắc Mỹ, Châu Âu và Trung Quốc. Ford cho biết việc sử dụng Android Automotive sẽ giúp tăng tính năng đa phương tiện và cải thiện trải nghiệm người dùng trên các dòng xe của hãng.*

-   **Tháng 9/2021**: *Honda vừa đưa ra thông báo sẽ tích hợp hệ điều hành Android Automotive trên tất cả các sản phẩm xe hơi từ năm 2022. Cho phép người dùng truy cập vào các tính năng thông minh và tiện ích của Google trên màn hình điều khiển trung tâm. Điều này cho thấy sự quan tâm của Honda đối với công nghệ và trải nghiệm người dùng trên các mẫu xe của họ. Việc tích hợp Android Automotive sẽ giúp tăng tính năng đa phương tiện và cải thiện trải nghiệm người dùng trên các mẫu xe của Honda. Các chi tiết về việc tích hợp hệ điều hành này trên các mẫu xe của Honda vẫn chưa được công bố.*

- **Từ tháng 3/2023:** *BMW sẽ tích hợp Android Automotive vào hệ điều hành BMW 8 cho một số mẫu xe mới, bao gồm cả dòng xe hơi và SUV. Việc tích hợp hệ điều hành này được coi là một bước đột phá trong việc nâng cao trải nghiệm người dùng và tăng tính năng đa phương tiện trên các mẫu xe của BMW.*

#### Android Automotive có gì khác so với Android Auto? Lựa chọn nào tốt hơn ?

* *Android Auto* là nền tảng chạy trên điện thoại của người dùng, đưa trải nghiệm người dùng Android Auto lên hệ thống thông tin giải trí trong xe tương thích qua kết nối USB. Android Auto hỗ trợ các ứng dụng được thiết kế để sử dụng trên xe.
* *Android Automotive* là một hệ điều hành và nền tảng chạy trực tiếp trên phần cứng của xe. Nó là một nền tảng full-stack, mã nguồn mở, có khả năng tùy biến cao, cung cấp năng lượng cho trải nghiệm thông tin giải trí. Android Automotive hỗ trợ các ứng dụng được xây dựng cho Android cũng như các ứng dụng được xây dựng cho Android Auto.

Cả Android Auto và Android Automotive đều là những hệ thống thông tin giải trí hàng đầu, giúp việc lái xe dễ dàng hơn. Tuy nhiên, Android Automotive được tinh chỉnh để điều khiển các thiết lập trên phương tiện, ví dụ như hệ thống điều hòa không khí, các chức năng âm thanh, và khóa cửa. Bạn có thể dễ dàng ra lệnh cho giao diện Android Automotive là “Tôi lạnh quá”, và nó sẽ điều khiển các thiết lập điều hòa để không khí trong xe trở nên ấm áp hơn.

Còn Android Auto chỉ giới hạn trong các chức năng cơ bản vốn không được tích hợp vào hệ thống của xe, như gọi điện thoại rảnh tay, cập nhật thời tiết, chơi nhạc, hoặc điều hướng. Android Automotive bao gồm những chức năng tương tự nhưng có độ chính xác cao hơn. Ví dụ, nếu đang lái xe điện, Android Automotive sẽ cung cấp tính năng dẫn đường GPS và tính toán chính xác quãng đường đi được cho đến khi hết pin.

Bên cạnh đó, bạn không phải lo điện thoại hết pin nếu dùng Android Automotive. Chưa kể, hầu hết các hệ thống thông tin giải trí sử dụng Android Automotive OS còn hỗ trợ cả Apple CarPlay và Android Auto.

Tuy nhiên, số lượng các mẫu xe được tích hợp Android Automotive trên thị trường hiện nay là rất ít. Trong khi đó, có đến hơn 500 mẫu xe hỗ trợ Android Auto. Kể cả khi hệ thống thông tin giải trí của xe bạn không tương thích với Android Auto, bạn vẫn có thể thay thế bằng một hệ thống stereo của bên thứ ba có hỗ trợ Android Auto.

Theo trang CarParts, chi phí để tích hợp Android Auto vào xe nếu hệ thống hiện tại không tương thích sẽ dao động từ 200 – 600 USD. Và bạn cần kiểm tra để chắc chắn hệ thống stereo bên thứ ba tương thích Android Auto kia lắp vừa vào xe bạn trước khi mua.

Gần đây, một số nghiên cứu đã được tiến hành về Android Auto, chủ yếu tập trung vào phân tích tĩnh của các ứng dụng giải trí được cung cấp. Mối đe dọa chính xuất hiện khi triển khai Android Auto là sự phân tán tập trung bởi việc sử dụng giao diện người dùng kém chất lượng, có thể ảnh hưởng đến an toàn khi lái xe. Các giáo sư đã phân tích và tìm thấy lỗ hổng JavaScript trong một phần từ tất cả các ứng dụng được phân tích, có thể được vá qua các bản cập nhật qua mạng (OTA) trong tương lai và do đó có thể giới hạn thiệt hại nghiêm trọng trong thời gian ngắn. Hiện nay, chỉ có vài loại xe ngoài Tesla cung cấp quy trình cập nhật OTA hoàn toàn. Điều này một phần là do tích hợp cồng kềnh của các giải pháp cập nhật hiện có vào kiến ​​trúc hệ thống đã cũ. Nếu một nhà sản xuất xe (OEM) cung cấp nhiều IVI trên các mẫu xe khác nhau với các hệ điều hành khác nhau, họ phải lặp lại quá trình tích hợp này, điều này có thể trở nên rất khó khăn. Android Automotive dựa trên Android và hỗ trợ cập nhật OTA trong nhiều năm qua. Dự án mã nguồn mở Android (AOSP) thậm chí đã đưa ra các hướng dẫn cho trường hợp sử dụng ô tô cụ thể, có "một số khác biệt nhỏ trong bước Tải xuống do hỗ trợ cho chế độ Garage". Bên cạnh việc cập nhật hệ điều hành IVI, các bản cập nhật firmware OTA cho các ECU khác cũng được hỗ trợ theo tài liệu của Google. "Chế độ Garage" được đề cập ở trên là chế độ tiết kiệm năng lượng mà một chiếc xe đỗ có thể sử dụng khi khởi động bị tắt. Điều này cho phép cập nhật OTA qua đêm khi xe đỗ trong garages và kết nối với mạng WiFi của khách hàng tại nhà. Mặc dù Google cho phép các OEM kích hoạt cập nhật USB trên IVI, nhưng chúng tôi rất khuyến khích chống lại quy trình cập nhật vật lý này do tăng nguy cơ an ninh, như đã được mô tả trong quá khứ. Chúng ta có thể giả định rằng với việc triển khai dần dần của Android Automotive trên các hệ thống IVI, quy trình cập nhật OTA có thể được tiêu chuẩn hóa và dễ dàng tích hợp hơn cho các nhà sản xuất ô tô trên tất cả các mẫu xe trong tương lai của họ.

#### Android Automotive sẽ là tương lai của hệ thống giải trí trên xe sau này

Có nhiều lợi ích của việc sử dụng Android Automotive trong hệ thống giải trí trên xe. Trước hết, Android Automotive sẽ mang lại trải nghiệm tương tự như sử dụng smartphone, cho phép người dùng tùy chỉnh và điều khiển các tính năng giải trí trên xe một cách dễ dàng. Ngoài ra, việc tích hợp các dịch vụ của Google như Google Maps và Google Assistant cũng giúp người dùng dễ dàng điều hướng và tương tác với hệ thống giải trí trên xe.

Hơn nữa, Android Automotive cũng cung cấp cho các hãng sản xuất ô tô một nền tảng phát triển mở, cho phép họ tùy chỉnh và phát triển các tính năng giải trí trên xe theo nhu cầu của từng mẫu xe và thị trường khác nhau. Điều này giúp tăng tính đa dạng và độc đáo cho các sản phẩm ô tô, đồng thời giúp các hãng sản xuất tiết kiệm được chi phí phát triển hệ thống riêng của mình.

Liên minh Open Automotive là nơi mà nhiều nhà sản xuất ô tô ngày nay hướng đến, họ cam kết sẽ tích hợp nền tảng Android vào các phương tiện di chuyển. Có nghĩa là tương lai của Android Auto lẫn Android Automotive sẽ khá tươi sáng.

Mặc dù hiện tại Android Automotive vẫn chưa phổ biến như Android Auto, nhưng với sự hợp tác của nhiều nhà sản xuất ô tô thông qua Liên minh Open Automotive có lẽ sẽ sớm thay đổi, xét việc ngày càng nhiều xe sắp ra mắt thị trường với hệ thống thông tin giải trí Android Automotive trong vài năm tới đây. Hệ thống này còn mang lại cho các nhà sản xuất một ưu thế, bởi họ có thể tùy biến giao diện mà không phải dựa vào các ứng dụng Google – một điều rất tốt với những ai xem trọng quyền riêng tư. Android Automotive là một bước đột phá trong ngành công nghiệp ô tô khi mang đến một hệ thống giải trí được tích hợp hoàn chỉnh và đầy đủ chức năng, đồng thời cũng mang lại khả năng tùy biến và linh hoạt cho các nhà sản xuất. Với sự phát triển của công nghệ và sự yêu cầu ngày càng cao về tiện ích và trải nghiệm của người dùng, Android Automotive được kỳ vọng sẽ trở thành tương lai của hệ thống giải trí trên xe.

Hơn nữa, Android Automotive còn được xem là một bước tiến đáng kể trong lĩnh vực xe tự lái và kết nối internet của vạn vật (IoT). Với các tính năng như điều khiển giọng nói và hỗ trợ kết nối 5G, Android Automotive có thể giúp tài xế và hành khách kết nối với internet và các thiết bị khác ngoài xe hơi một cách nhanh chóng và tiện lợi.

Tuy nhiên, điều quan trọng nhất là đảm bảo an toàn và bảo mật trong khi sử dụng Android Automotive. Với sự phát triển nhanh chóng của công nghệ và ngày càng nhiều thiết bị được kết nối với internet, việc bảo vệ các thiết bị khỏi các cuộc tấn công trở nên quan trọng hơn bao giờ hết.

Android Automotive cũng cho phép các nhà sản xuất ô tô phát triển các tính năng và ứng dụng độc quyền dành riêng cho phương tiện của mình. Điều này mang lại sự tùy biến cao hơn và giúp tạo ra trải nghiệm người dùng tốt hơn. Với sự phát triển của công nghệ xe tự lái, hệ thống giải trí trên xe sẽ trở thành một phần quan trọng trong trải nghiệm lái xe tự động. Android Automotive đang chuẩn bị để đáp ứng nhu cầu này, cung cấp các tính năng giải trí và thông tin cần thiết để giúp tài xế và hành khách thư giãn và giải trí trong khi xe tự động lái.

Do đó, các nhà sản xuất ô tô và nhà phát triển phải đảm bảo rằng họ có các biện pháp bảo mật và kiểm soát nghiêm ngặt để bảo vệ các hệ thống xe hơi của mình khỏi các cuộc tấn công và truy cập trái phép.

Tóm lại, Android Automotive đang trở thành xu hướng trong lĩnh vực xe hơi thông minh và giải trí trên xe, trở thành một phần quan trọng trong việc tạo ra trải nghiệm lái xe tốt hơn và an toàn hơn. Nó đại diện cho sự tiến bộ trong hệ thống giải trí trên xe và đang được các nhà sản xuất ô tô và người dùng trên toàn thế giới đón nhận và kỳ vọng. Với sự phát triển nhanh chóng của công nghệ, việc đảm bảo an toàn và bảo mật trong khi sử dụng hệ thống này trở nên cực kỳ quan trọng. Tuy nhiên, với những tiềm năng và ưu thế của nó, không có nghi ngờ gì về việc Android Automotive sẽ là tương lai của hệ thống giải trí trên xe.

### Tổng quan về vấn đề tìm bãi đỗ xe

#### Thực trạng số lượng xe ô tô và số lượng bãi đỗ hiện nay

Hiện nay, các thành phố lớn của Việt Nam đang chịu sức ép từ việc gia tăng nhanh chóng của các loại phương tiện tham gia giao thông với tỷ lệ tăng hàng năm từ 8 đến 13% và cùng với việc tăng dân số đô thị do quá trình đô thị hóa diễn ra nhanh chóng . Trong khi đó, sự phát triển của hệ thống kết cấu hạ tầng giao thông đô thị (GTĐT) vẫn không đáp ứng đủ so với nhu cầu thực tế, đặc biệt là quỹ đất dành cho các điểm đỗ xe, bãi đỗ xe còn rất thấp. Diện tích bãi đỗ xe của Hà Nội và thành phố Hồ Chí Minh hiện nay lần lượt là 0,3 và 0,8%, trong khi yêu cầu từ 3 đến 5% trên tổng diện tích đất đô thị .

Nhu cầu đỗ xe ở các khu vực đô thị là rất cao với kết quả khảo sát cho thấy trung bình có khoảng 95% thời gian của xe ô tô là đỗ và còn lại là đi tìm chỗ đỗ hoặc đang vận hành.

Hensher và King đã thực hiện nghiên cứu các điểm đỗ xe khác nhau ở trong trung tâm thành phố Sydney và cho thấy nếu áp dụng giải pháp PDM thông qua việc tăng 10% phí đỗ xe sẽ dẫn đến giảm 5,41% nhu cầu xe đi tới trung tâm, tăng 3,63% số chuyển sử dụng điểm đỗ và đi (park & ride, P&R), tăng 2,91% số chuyến đi sử dụng GTCC và giảm 4,69% số chuyến đi đến trung tâm thành phố. Ngoài ra, việc áp dụng giải pháp PDM kết hợp với các dịch vụ giao thông thông minh (GTTM) đã phát huy được hiệu quả cao trong quản lý và vận hành giao thông đô thị.

Hiện nay, tình trạng thiếu hụt bãi đỗ xe là một vấn đề lớn trong giao thông đô thị tại các thành phố lớn của Việt Nam. Đây là một trong những nguyên nhân chính dẫn đến tình trạng kẹt xe, ùn tắc giao thông và tốn nhiều thời gian di chuyển trong các giờ cao điểm. Dưới đây là một số minh chứng cụ thể về thực trạng bãi đỗ xe tại một số thành phố lớn của Việt Nam:

* Tại Hà Nội, diện tích bãi đỗ xe của thành phố là rất thấp, chỉ chiếm khoảng 0,3% tổng diện tích đất đô thị. Các bãi đỗ xe công cộng hiện nay cũng không đáp ứng được nhu cầu đỗ xe của người dân và du khách, đặc biệt là trong các khu vực trung tâm.
* Ở TP.HCM, diện tích bãi đỗ xe cũng khá hạn chế, chỉ khoảng 0,8% tổng diện tích đất đô thị. Nhiều khu vực trong thành phố không có đủ chỗ đỗ xe, dẫn đến tình trạng xe đỗ lấn chiếm vỉa hè, dẫn đến tình trạng kẹt xe và cản trở cho người đi bộ.
* Tại các thành phố khác như Đà Nẵng, Hải Phòng, Nha Trang, Vũng Tàu, tình trạng thiếu hụt bãi đỗ xe cũng tương tự. Các bãi đỗ xe công cộng chủ yếu tập trung ở khu vực trung tâm, trong khi nhu cầu đỗ xe của người dân và du khách ở các khu vực phía ngoại thành và ven biển cũng rất lớn.
* Một số bãi đỗ xe tư nhân cũng đã lấn chiếm vỉa hè, đường phố hoặc các khu vực công cộng để tăng diện tích đỗ xe, gây cản trở cho người đi bộ và ảnh hưởng đến mỹ quan thành phố.

Trên đây là những minh chứng cụ thể cho thấy tình trạng thiếu hụt bãi đỗ xe là một vấn đề lớn trong giao thông đô thị tại Việt Nam, gây khó khăn cho việc di chuyển và ảnh hưởng đến sự phát triển của đô thị.

Theo báo cáo của Bộ Giao thông Vận tải, năm 2015 cả nước có hơn hai triệu ô tô đang lưu hành, năm 2016 là 2.5triệu chiếc, số lượng xe từ 9 chỗ trở xuống chiếm gần 50%.

Bảng 1-1 Số lượng ô tô đang lưu hành năm 2015-2016 (nguồn: Bộ GTVT)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Số lượng | | Tỷ lệ | |
| Các loại xe | 2015 | 2016 | 2015 | 2016 |
| Tổng số xe các loại | 2101401 | 2516144 | 100% | 100% |
| 1. Xe con từ 9 chỗ trở xuống | 1033131 | 1270066 | 49% | 50% |
| 2. Xe khách từ 9 chỗ trở lên | 118030 | 136727 | 6% | 5% |
| Từ 10 đến 25 chỗ | 66277 | 74987 |  |  |
| Từ 26 đến 46 chỗ | 37874 | 45451 |  |  |
| Trên 46 chỗ | 13879 | 16289 |  |  |
| 3. Xe tải | 853735 | 1004499 | 41% | 40% |
| Tải trọng đến 2 tấn | 444548 | 516010 |  |  |
| Tải trọng từ trên 2 tấn đến 7 tấn | 226336 | 270080 |  |  |
| Tải trọng từ trên 7 tấn đến 20 tấn | 177967 | 212159 |  |  |
| Tải trọng trên 20 tấn | 4864 | 6250 |  |  |
| 4. Xe chuyên dùng và xe khác | 96505 | 104852 | 5% | 4% |

Quyết định số 356/QĐ-TTg 25/2/2013 của Thủ tướng chính phủ “Phê duyệt điều chỉnh Quy hoạch phát triển giao thông vận tải đường bộ Việt Nam đến năm 2020 và định hướng đến năm 2030” chỉ rõ: đến năm 2020 cả nước có 3.2-3.5 triệu ô tô, 57% ô tô con; 14% xe khách, 29% xe tải. Với tốc độ phát triển của nền kinh tế trong thời kỳ hội nhập như hiện nay, mục tiêu quy hoạch số lượng ô tô sẽ sớm đạt được, nhất là trong bối cảnh Việt Nam đang mở cửa thị trường ô tô nhập khẩu với các nước ASEAN và tiếp theo sẽ là các thị trường sản xuất ô tô của các nước phát triển khác ở khu vực Châu Á Thái Bình Dương, Châu Âu, Châu Mỹ.v.v.

Số lượng phương tiện giao thông tăng cao sẽ thúc đẩy phát triển kinh tế, nhưng mặt khác sẽ gây áp lực rất lớn đến hạ tầng giao thông, đặc biệt là giao thông đô thị ở Việt Nam vốn đã phát sinh rất nhiều vấn đề phức tạp. Các vấn đề về hệ thống giao thông tĩnh, đặc biệt là vấn đề đỗ xe trong đô thị, sẽ là bài toán tiếp theo mà ngành giao thông đô thị phải đối mặt và tìm giải pháp thích hợp.

- Quy hoạch và xây dựng bãi đỗ xe trong đô thị Việt Nam

Vấn đề quy hoạch giao thông tĩnh và bãi đỗ xe đã được đề cập khá nhiều trong các bản quy hoạch chung và quy hoạch chi tiết ở các thành phố lớn của Việt Nam như:

Luật Giao thông đường bộ 2008 Điều 51, khoản 1: “Trong đô thị, khi xây dựng trụ sở cơ quan, trường học, bệnh viện, trung tâm thương mại - dịch vụ, văn hóa và khu dân cư phải xây dựng đủ nơi đỗ xe phù hợp với quy mô của công trình.”

Quyết định 519/QĐ-TTg của Thủ tướng chính phủ ngày 31/3/2016 “Phê duyệt Quy hoạch giao thông vận tải Thủ đô Hà Nội đến năm 2030, tầm nhìn 2050” nêu rõ “Diện tích đất dành cho giao thông 20-26% đối với đô thị trung tâm, 18-23% đối với đô thị vệ tinh, 16-20% đối với các thị trấn. Trong đó, diện tích đất dành cho giao thông tĩnh cần đạt 3-4%”

Quyết định 568/QĐ-TTg của Thủ tướng chính phủ ngày 08/04/2013 “Phê duyệt Điều chỉnh Quy hoạch phát triển giao thông vận tải thành phố Hồ Chí Minh đến năm 2020 và tầm nhìn sau năm 2020”, Thành phố Hồ Chí Minh cần 1146 ha dành cho các bến bãi.

Quy hoạch tổng thể hệ thống giao thông tĩnh thành phố Đà Nẵng đến năm 2020, tầm nhìn đến năm 2030, quyết định 5216/QĐ-UBND ngày 21/7/2015 của chủ tịch UBND TP Đà Nẵng: đến năm 2020 có 131ha và đến 2030 có 322.5ha.

Quyết định 1448/QĐ-TTg ngày 16 tháng 09 năm 2009 của Thủ tướng chính phủ, “Phê duyệt điều chỉnh Quy hoạch chung xây dựng thành phố Hải Phòng đến năm 2025 và tầm nhìn đến năm 2050”, diện tích dành cho bãi đỗ xe cần phải có là 436.29 ha

Quy hoạch chi tiết mạng lưới bến bãi đỗ xe công cộng trên địa bàn thành phố Cần Thơ đến năm 2030, 209/QĐ-UBND, 25 tháng 01 năm 2017: Dành 123.5 ha, chiếm 3.3% đất dành cho giao thông (2021-2030); 77.1 ha, chiếm 3% đất dành cho giao thông (2016-2020)

Như vậy, vấn đề về quy hoạch bãi đỗ xe đã có các văn bản pháp quy điều chỉnh, xong công tác xây dựng, quản lý bãi đỗ xe hiện nay vẫn chưa theo kịp nhu cầu phát triển. Việc triển khai xây dựng hệ thống giao thông tĩnh ở các thành phố lớn của Việt Nam vẫn đang gặp các vấn đề khó khăn nhất định.

Hiện nay, thành phố Hồ Chí Minh và Hà Nội, việc thiếu hụt bãi đỗ xe được đánh giá là khá nghiêm trọng, các cụm từ xuất hiện nhiều trên các kênh thông tin chính thống khi nói về bãi đỗ xe hiện nay ở Việt Nam: “thiếu hụt trầm trọng”, “vỡ quy hoạch”, “vừa thiếu”, “vừa kém”.v.v..

Đối với thành phố Hồ Chính Minh, “Theo TS Phạm Sanh - chuyên gia giao thông, với lượng ô tô khoảng 1 triệu chiếc, TP cần trên 1.000 ha để đậu xe. Hiện TP mới có khoảng 200 ha được sử dụng làm các bến, bãi đậu xe, thiếu khoảng 800 ha. Đặc biệt, riêng khu vực trung tâm cần khoảng 200 - 300 ha” .

Đối với thành phố Hà Nội, “Hiện nay, trên địa bàn thành phố diện tích đất dành cho giao thông tĩnh và các điểm, bãi đỗ xe công cộng mới chỉ đáp ứng được 8-10% nhu cầu, còn lại 90-92% số phương tiện đang đỗ tại các điểm đỗ xe của chung cư, khu đô thị, sân cơ quan, công sở, lòng đường, sân trường, bệnh viện, hoặc tại các khu đất trống chờ dự án” .

Tại thành phố Đà Nẵng, hiện nay, vấn đề thiếu hụt bãi đỗ xe đã bắt đầu được chú ý, “Đà Nẵng có 17 bãi đỗ ô tô đã và đang đầu tư, không đáp ứng kịp đà tăng nhanh các phương tiện giao thông hiện nay” .

Rõ ràng, vấn đề giao thông tĩnh và bãi đỗ xe là vấn đề tiếp theo cần được quan tâm trong hoạt động giao thông đô thị Việt Nam.

Quá trình di chuyển tìm kiếm bãi đỗ xe được chứng minh là rất phức tạp, tiêu tốn nhiều thời gian, chi phí và các vấn đề liên quan khác như tai nạn giao thông, tiêu hao nhiên liệu, khí thải. Vì vậy, ở các nước phát triển, lượng ô tô tham gia giao thông lớn, các chính sách quản lý bãi đỗ, cung cấp thông tin đã được phát triển từ rất sớm.

#### Những hậu quả mà sự chênh lệch số xe và bãi đỗ để lại.

Tình trạng các loại phương tiện tham gia giao thông gia tăng nhanh chóng dẫn đến diện tích lòng đường và vỉa hè bị lấn chiếm để đỗ xe, sử dụng sai mục đích hoặc đỗ xe tại các điểm trái phép. Hậu quả là dẫn đến tình trạng ùn tắc giao thông càng thêm trầm trọng tại các khu vực trung tâm đô thị, ảnh hưởng tiêu cực đến cuộc sống và mỹ quan đô thị.

Các thành phố lớn hiện nay đang đối mặt với sự thiếu hụt diện tích bãi đỗ xe so với nhu cầu và đồng thời không thể thực hiện được giải pháp mở  rộng diện tích bãi đỗ do vấn đề về chi phí mặt bằng, đặc biệt ở các khu vực trung tâm giao dịch có tập trung nhiều hoạt động kinh doanh, mua bán, vui chơi và giải trí. Kinh nghiệm của các nước phát triển trên thế giới đã chỉ ra sự cần thiết của việc áp dụng những giải pháp quản lý nhu cầu đỗ xe (parking demand management, PDM) để giảm sự tập trung của nhu cầu đỗ xe thông qua việc hạn chế sử dụng phương tiện cơ giới cá nhân và đồng thời chuyển dịch sang sử dụng phương thức giao thông công cộng (GTCC) sẵn có ở trong các khu vực đô thị, mặt khác đáp ứng hợp lý nhu cầu đỗ xe ở những khu vực xung quanh .

Ngoài ra, sự chênh lệch số xe và bãi đỗ cũng góp phần vào tình trạng ô nhiễm môi trường và khí hậu. Việc tăng lượng xe cơ giới không chỉ dẫn đến ùn tắc giao thông mà còn tăng lượng khí thải, làm ảnh hưởng đến chất lượng không khí và góp phần vào hiện tượng biến đổi khí hậu.

Sự chênh lệch số xe và bãi đỗ cũng ảnh hưởng đến đời sống xã hội, khiến cho một số người cảm thấy khó chịu và bức xúc khi phải điều khiển xe cộ trong tình trạng ùn tắc. Đồng thời, nó cũng góp phần vào tình trạng tai nạn giao thông, đặc biệt là các tai nạn do vi phạm quy định về đỗ xe.

Hơn nữa, sự chênh lệch số xe và bãi đỗ cũng ảnh hưởng đến kinh tế và tài chính của các thành phố. Việc mở rộng diện tích bãi đỗ sẽ đòi hỏi chi phí đầu tư khá lớn, đặc biệt là ở các khu vực trung tâm đô thị. Tuy nhiên, không giải quyết được vấn đề này sẽ dẫn đến tình trạng ùn tắc giao thông, tăng chi phí hoạt động và giảm hiệu quả kinh doanh của các doanh nghiệp. Ngoài ra, việc sử dụng các điểm đỗ xe không chính thống cũng sẽ tạo ra các khoản chi phí không đáng có cho người dân.

Hiện tại, trên thế giới đã có nhiều nghiên cứu về tác động của sự chênh lệch số lượng xe và bãi đỗ đến môi trường và giao thông. Dưới đây là một số minh chứng được đưa ra:

* Nghiên cứu tại Thành phố New York, Mỹ cho thấy, sự chênh lệch số lượng xe và bãi đỗ đã góp phần làm tăng lượng khí thải độc hại và khói bụi, gây ảnh hưởng đến sức khỏe của người dân trong thành phố.
* Tại thành phố Amsterdam, Hà Lan, sự chênh lệch giữa số lượng xe và bãi đỗ đã làm tăng tắc đường và thời gian di chuyển. Theo một nghiên cứu của Trường Đại học Amsterdam, việc giảm 10% số lượng xe trên đường có thể làm giảm tắc đường và thời gian di chuyển đến 25%.
* Tại Thành phố Singapore, chính phủ đã thực hiện các chính sách giới hạn số lượng xe đăng ký để giảm sự chênh lệch giữa số lượng xe và bãi đỗ. Kết quả là, tình trạng kẹt xe và ùn tắc giao thông đã giảm đáng kể trong thành phố này.
* Tại Trung Quốc, một nghiên cứu được thực hiện tại thành phố Thượng Hải cho thấy, sự chênh lệch giữa số lượng xe và bãi đỗ đã góp phần làm tăng lượng khí thải và gây ảnh hưởng đến chất lượng không khí trong thành phố.
* New York, Hoa Kỳ: Theo một nghiên cứu của NYC Department of Transportation, trong vòng 15 năm qua, số lượng xe ở Manhattan đã tăng 44% trong khi diện tích bãi đỗ chỉ tăng 1%. Kết quả là, người dân và du khách gặp khó khăn khi tìm kiếm chỗ đỗ xe và thường phải đi lại quá xa từ chỗ đỗ xe đến địa điểm đến. Điều này cũng gây ra ùn tắc giao thông và tác động đến chất lượng không khí.
* Tokyo, Nhật Bản: Tokyo là một trong những thành phố đông dân nhất trên thế giới, với hơn 37 triệu người sinh sống trong vùng đô thị của nó. Số lượng xe tăng đột biến trong những năm qua, tuy nhiên diện tích bãi đỗ không tăng tương ứng. Kết quả là, người dân thường phải chờ đợi để tìm kiếm chỗ đỗ xe, gây ra tình trạng ùn tắc và tác động đến chất lượng không khí.
* Mumbai, Ấn Độ: Đây là một trong những thành phố phát triển nhanh nhất trên thế giới, với hơn 20 triệu dân sống trong vùng đô thị của nó. Tuy nhiên, diện tích bãi đỗ xe tại Mumbai rất hạn chế, khiến việc tìm chỗ đỗ xe trở nên cực kỳ khó khăn. Số lượng xe tăng lên theo thời gian đã tạo ra sự cạnh tranh lớn về chỗ đỗ xe, đặc biệt là trong khu vực trung tâm của thành phố.

# PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG



### Xác định yêu cầu

* Đối với Admin:

Theo dõi, xử lý việc đặt bãi đỗ, quá trình quá trình đặt bãi đỗ.

Xem và tra cứu thông tin đơn đặt.

Xem, theo dõi bãi đỗ, đơn đặt và khách hàng.

* Đối với Khách hàng:

Dễ hiểu, dễ dùng, hấp dẫn, dễ tìm kiếm thông tin bãi đỗ.

Đảm bảo an toàn tuyệt đối thông tin của khách hàng.

### Đưa ra các chức năng của hệ thống

* Backend

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Chức năng | Người dùng | Mô tả |
| 1 | Đăng nhập | Admin | Chức năng cho phép admin đăng nhập vào hệ thống |
| 2 | Bảo trì vị trí các bãi đỗ xe ở gần | Admin | Chức năng cho phép admin xem, thêm, sửa, xóa bảng BaiDoXe |
| 3 | Bảo trì Đơn đặt | Admin | Chức năng cho phép admin xem, hủy bảng GiaoDich |
| 4 | Quản lý địa điểm. | Admin | Chức năng cho phép admin xem, thêm, xóa trong bảng GiaoDich |
| 5 | Xác nhận đơn đặt. | Admin | Chức năng cho phép admin xác nhận đơn hàng khách hàng đã đặt |
| 6 | Quản lý người dung | Admin | Chức năng cho phép admin xem, thêm, sửa, xóa bảng NguoiDung |

* Frontend

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Chức năng | Người dùng | Mô tả |
| 1 | Đăng kí | Khách hàng | Chức năng cho phép khách hàng đăng kí tài khoản |
| 2 | Đăng nhập | Khách hàng | Chức năng cho phép khách hàng đăng nhập vào hệ thống |
| 3 | Xem vị trí các bãi đỗ xe ở gần | Khách hàng | Chức năng cho phép khách hàng xem các vị trí có bãi đỗ xe ở gần đó. |
| 4 | Đặt vị trí đỗ xe | Khách hàng | Chức năng cho phép khách hàng đặt vị trí đỗ xe. |
| 5 | Hủy đơn đặt bãi đỗ | Khách hàng | Chức năng cho phép khách hàng hủy đơn đã đặt |
| 6 | Thay đổi thông tin cá nhân | Khách hàng | Chức năng cho phép khách hàng thay đổi thông tin cá nhân. |
| 7 | Đánh giá | Khách hàng | Chức năng cho phép khách hàng đánh giá về dịch vụ bãi đỗ |

### Sơ đồ Use case

* Backend

A picture containing diagram, line, circle

Description automatically generated

Hình 2‑1: Sơ đồ Use case phía backend

* FrontEnd

A picture containing diagram, line, circle, text

Description automatically generated

Hình 2‑2: Sơ đồ Use case phía FrontEnd

### Phân tích dặc tả Use case

* **Mô tả use case đăng kí**
* Tên use case: đăng kí
* Luồng cơ bản:

+ Use case bắt đầu khi khách hàng kích vào nút “ Đăng kí” ở trang chủ. Hệ thống sẽ hiển thị form Đăng kí lên màn hình.

+ Khách hàng nhập đầy đủ thông tin theo hướng dẫn, sau đó kích vào nút “Đăng ký”. Hệ thống sẽ kiểm tra và lưu thông tin khách hàng vào bảng USERS.

+ Use case kết thúc

* Luồng rẽ nhánh:

+ Tại bất kì bước nào của quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hiển thị thông báo lỗi và kết thúc use case.

+ Tại bước 2 trong luồng cơ bản, nếu khách không nhập đầy đủ thông tin, hệ thống thông báo thiếu thông tin cần nhập và yêu cầu khách hàng nhập lại

+ Tại bước 2 trong luồng cơ bản, tên tài khoản trùng với tên tài khoản đã tồn tại, hệ thống sẽ đưa ra thông báo “ Tên tài khoản đã tồn tại” và yêu cầu khách hàng nhập lại.

* Điểm mở rộng: Không có
* Tiền điều kiện: Không có
* Hậu điều kiện: Không có
* Yêu cầu đặc biệt: Không có
* **Mô tả use case đăng nhập**
* Use case cho phép người dùng đăng nhập
* Tên use case: Đăng nhập
* Luồng cơ bản:

+ Use case bắt đầu khi khách hàng kích vào nút “Đăng nhập” trên trang chủ. Hệ thống sẽ hiển thị form đăng nhập lên màn hình.

+ Khách hàng nhập email và mật khẩu sau đó kích nút “Đăng nhập”. Hệ thống sẽ kiểm tra tên tài khoản và mật khẩu và đưa khách hàng đến trang chủ.

+ Use case kết thúc.

* Luồng rẽ nhánh:

+ Tại bước 2 của luồng cơ bản, nếu khách hàng nhập sai tên tài khoản hoặc mật khẩu, hệ thống sẽ hiển thị thông báo ”Tên đăng nhập hoặc mật khẩu không đúng!” và yêu cầu nhập lại.

+ Tại bước 2 của luồng cơ bản, nếu khách hàng nhập đúng tên tài khoản đang bị khóa, hệ thống “Tài khoản bạn đang bị khóa hoặc chưa được xác nhận. Vui lòng liên hệ với ADMIN để được xử lý!” và yêu cầu nhập lại.

+ Tại bất kì bước nào trong quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hiển thị thông báo lỗi và kết thúc use case.

* Điểm mở rộng: Không có
* Tiền điều kiện: Không có
* Hậu điều kiện: Không có
* Yêu cầu đặc biệt: Không có
* **Mô tả use case xem vị trí bãi đỗ**
* Use case cho phép người dùng có thể xem các vị trí có bãi đỗ xe ở gần đó
* Tên use case: Xem vị trí bãi đỗ
* Luồng cơ bản:

+ Use case bắt đầu khi khách hàng kích vào nút “Xem vị trí bãi đỗ” trên trang chủ. Hệ thống sẽ hiển thị bản đồ kèm theo các bãi đỗ xe gần nhất lên màn hình.

+ Người dùng có thể kéo và thả bản đồ để xem các vị trí đỗ khác. Người dùng chọn một trong số các bãi đỗ xe đã hiển thị để xem thông tin chi tiết về bãi đỗ xe đó. Hệ thống hiển thị thông tin của bãi đỗ xe lấy từ CSDL bao gồm tên, địa chỉ, sức chứa, giá và các thông tin khác liên quan.

+ Người dùng có thể dóng trang thông tin chi tiết bãi đỗ xe để quay lại trang bản đồ và xem các bãi đỗ xe khác.

+ Use case kết thúc.

* Luồng rẽ nhánh:

+ Tại bước 1 của luồng cơ bản, nếu không có kết nối internet, hệ thống sẽ thông báo lỗi và yêu cầu người dùng kết nối internet để sử dụng tính năng này.

+ Tại bước 2 của luồng cơ bản, Nếu hệ thống không tìm thấy bãi đỗ nào trong khu vực quanh vị trí hiện tại của người dùng hoặc địa điểm được chọn, hệ thống sẽ thông báo và đưa ra lời khuyên cho người dùng di chuyển đến khu vực khác hoặc thử lại sau.

+ Tại bước 2 của luồng cơ bản nếu người dùng không thể tìm kiếm được bãi đỗ xe nào trên bản đồ hoặc thông tin của bãi đỗ xe không đầy đủ hoặc sai, người dùng có thể báo cáo lỗi cho hệ thống để cải thiện chất lượng dịch vụ.

+ Tại bất kì bước nào trong quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hiển thị thông báo lỗi và kết thúc use case.

* Điểm mở rộng: Không có
* Tiền điều kiện: Hệ thống đã có dữ liệu về vị trí các bãi đỗ xe và có kết nối internet
* Hậu điều kiện: Không có
* Yêu cầu đặc biệt: Không có
* **Mô tả use case đặt vị trí đỗ xe**
* Use case cho phép người dùng đặt vị trí đỗ xe
* Tên use case: Đặt vị trí đỗ xe
* Luồng cơ bản:

+ Use case bắt đầu khi người dùng kích vào nút “Đặt vị trí đỗ xe” trên trang chủ. Hệ thống hiển thị bản đồ kèm danh sách các bãi đỗ xe có sẵn và gần địa điểm mà người dùng muốn đến.

+ Người dùng chọn bãi đỗ xe mà mình muốn đặt chỗ. Hệ thống hiển thị thông tin chi tiết về bãi đỗ xe bao gồm tên, địa chỉ, số lượng chỗ đỗ, giá cả và thời gian còn lại cho việc đặt chỗ.

+ Người dùng chọn số lượng xe và thời gian đặt chỗ. Người dùng xác nhận thông tin. Hệ thống gửi thông tin xác nhận đặt chỗ thành công đến người dùng.

+ Người dùng tới bãi đỗ xe đưa mã xác nhận để xác nhận việc đặt chỗ của mình. Hệ thống xác nhận địa chỉ của bãi đỗ xe, số chỗ xe đã đặt và thời gian đặt chỗ.

+ Use case kết thúc.

* Luồng rẽ nhánh:

+ Tại bước 1 của luồng cơ bản, nếu hệ thống không tìm thấy bãi đỗ xe nào gần địa điểm mà người dùng muốn đến, hệ thống sẽ thông báo lỗi cho người dùng và đưa ra lời khuyên cho người dùng tìm kiếm bãi đỗ khác.

+ Tại bước 3 của luồng cơ bản, nếu số lượng vị trí đỗ xe đã được đặt hết trước khi người dùng hoàn tất việc đặt chỗ, hệ thống sẽ thông báo lỗi cho người dùng và yêu cầu người dùng tìm chỗ đỗ khác.

+ Tại bước 4 của luồng cơ bản, nếu người dùng quá thời gian đặt chỗ, hệ thống sẽ hủy bỏ việc đặt chỗ của người dùng và thông báo cho người dùng biết.

+ Tại bất kì bước nào trong quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hiển thị thông báo lỗi và kết thúc use case.

* Điểm mở rộng: Không có
* Tiền điều kiện: Hệ thống đang hoạt động, Đã có tài khoản người dùng được đăng ký thành công trên ứng dụng, có kết nối internet.
* Hậu điều kiện: Không có
* Yêu cầu đặc biệt: Không có
* **Hủy đơn đặt bãi đỗ**
* Use case cho phép người dùng hủy chỗ đỗ tại bãi đỗ đã đặt
* Tên use case: Hủy đơn đặt bãi đỗ
* Luồng cơ bản:

+ Use case bắt đầu khi khách hàng kích vào nút “Hủy đơn đặt bãi đỗ” trên trang chủ. Hệ thống hiển thị danh sách các đơn đặt chỗ đã được đặt trước đó của người dùng.

+ Người dùng chọn đơn đặt bãi đỗ cần hủy. Hệ thống yêu cầu xác nhận

+ Người dùng xác nhận hành động và nhấn “Hủy đơn đặt bãi đỗ”. Hệ thống yêu cầu người dùng nhập lý do hủy đơn đặt bãi đỗ (tùy chọn).

+ Người dùng nhập lý do hủy đơn và xác nhận. Hệ thống thông báo cho người dùng biết đơn đặt chỗ đã được hủy thành công và gửi email hoặc tin nhắn xác nhận cho người dùng.

+ Use case kết thúc.

* Luồng rẽ nhánh:

+ Tại bước 1 của luồng cơ bản, nếu kết nối internet bị lỗi, hệ thống sẽ thông báo lỗi và yêu cầu người dùng kết nối internet để sử dụng tính năng này.

+ Tại bước 2 của luồng cơ bản, nếu đơn đặt chỗ đã được hủy trước đó bởi người dùng hoặc bởi hệ thống, hệ thống sẽ thông báo cho người dùng biết rằng đơn đặt chỗ đã bị hủy và không thể hủy lại.

+ Tại bất kì bước nào trong quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hiển thị thông báo lỗi và kết thúc use case.

* Điểm mở rộng: Không có
* Tiền điều kiện: Hệ thống đang hoạt động, Người dùng đã đăng nhập, có đơn đặt chỗ bãi đỗ.
* Hậu điều kiện: Không có
* Yêu cầu đặc biệt: Không có
* **Mô tả use case thay đổi thông tin cá nhân**
* Use case cho phép người dùng có thể cập nhật thông tin cá nhân của mình trên hệ thống
* Tên use case: Thay đổi thông tin cá nhân
* Luồng cơ bản:

+ Use case bắt đầu khi khách hàng kích vào nút “Thông tin cá nhân” trên trang chủ. Hệ thống hiển thị thông tin cá nhân của người dùng và các mục cần thay đổi.

+ Người dùng chọn mục cần thay đổi và nhập thông tin mới. Hệ thống yêu cầu xác nhận hành động và kiểm tra tính hợp lệ của thông tin mới.

+ Người dùng xác nhận hành động và lưu thông tin mới. Hệ thống hiển thị thông báo cho người dùng biết thông tin cá nhân đã được cập nhật thành công.

+ Use case kết thúc.

* Luồng rẽ nhánh:

+ Tại bước 1 của luồng cơ bản, Nếu kết nối internet bị lỗi, hệ thống sẽ thông báo lỗi và yêu cầu người dùng kết nối internet để sử dụng tính năng này.

+ Tại bước 2 của luồng cơ bản, Nếu thông tin nhập không hợp lệ, ví dụ như địa chỉ email không đúng định dạng hoặc mật khẩu không đủ mạnh, hệ thống sẽ thông báo cho người dùng biết và yêu cầu người dùng nhập lại thông tin.

+ Tại bất kì bước nào trong quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hiển thị thông báo lỗi và kết thúc use case.

* Điểm mở rộng: Không có
* Tiền điều kiện: Hệ thống đang hoạt động, người dùng đã đăng nhập
* Hậu điều kiện: Không có
* Yêu cầu đặc biệt: Không có
* **Mô tả use case đánh giá**
* Use case cho phép người dùng đánh giá chất lượng dịch vụ của hệ thống
* Tên use case: Đánh giá
* Luồng cơ bản:

+ Use case bắt đầu khi khách hàng kích vào nút “Đánh giá” trên trang chủ. Hệ thống sẽ hiển thị danh sách lần người dùng sử dụng dịch vụ.

+ Người dùng chọn dịch vụ cần đánh giá. Hệ thống yêu cầu người dùng nhập nội dung đánh giá và đánh giá số sao tương ứng.

+ Người dùng nhập nội dung đánh giá và đánh giá số sao tương ứng. Hệ thống Yêu cầu xác nhận hành động.

+ Người dùng xác nhận hành động đánh giá dịch vụ. Hệ thống hiển thị thông báo đánh giá đã được gửi thành công.

+ Use case kết thúc.

* Luồng rẽ nhánh:

+ Tại bước 1 của luồng cơ bản, nếu kết nối internet bị lỗi, hệ thống sẽ thông báo lỗi và yêu cầu người dùng kết nối internet để sử dụng tính năng này.

+ Tại bước 2 của luồng cơ bản, Nếu người dùng đã đánh giá sản phẩm hoặc dịch vụ trước đó, hệ thống sẽ thông báo cho người dùng biết rằng đã có đánh giá của người dùng và không thể đánh giá lại.

+ Tại bất kì bước nào trong quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hiển thị thông báo lỗi và kết thúc use case.

* Điểm mở rộng: Không có
* Tiền điều kiện: Hệ thống đang hoạt động, người dùng đã đăng nhập và truy cập vào mục đánh giá.Hậu điều kiện: Không có
* Yêu cầu đặc biệt: Không có

### Thiết kế cơ sở dữ liệu

* **Sơ đồ cơ sở dữ liệu**

A screenshot of a computer

Description automatically generated with low confidence

Hình 2‑3: Sơ đồ cơ sở dữ liệu của hệ thống

* **Khởi tạo cơ sở dữ liệu:**
* BaiDoXe

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Giải thích | Kiểu dữ liệu | Khóa |
| 1 | id | Mã bãi đỗ | Int(11) | Khóa chính |
| 2 | ten | Tên bãi đỗ | Varchar(255) |  |
| 3 | dia\_chi | địa chỉ của bãi đỗ xe | int |  |
| 4 | latitude | lưu trữ giá trị vĩ độ | Decimal(10,8) |  |
| 5 | longitude | lưu trữ giá trị kinh độ | Decimal(11,8) |  |
| 6 | suc\_chua | số lượng chỗ đỗ xe có sẵn trong bãi | Int(11) |  |
| 7 | gia | lưu trữ giá tiền để đỗ xe | Float |  |
| 8 | danh\_gia\_trung\_binh | điểm đánh giá trung bình của bãi đỗ xe | Float |  |
| 9 | so\_luot\_danh\_gia | số lượt người dùng đã đánh giá bãi đỗ xe | Int(11) |  |
| 10 | id\_tinh\_thanh | lưu mã tỉnh thành | Int(11) | Khóa ngoài |

* NguoiDung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Giải thích | Kiểu dữ liệu | Khóa |
| 1 | id | Mã người dùng | Int(11) | Khóa chính |
| 2 | ten | lưu trữ tên của người dùng | Varchar(255) |  |
| 3 | email | lưu trữ địa chỉ email của người dùng | Varchar(255) |  |
| 4 | mat\_khau | lưu trữ giá trị vĩ độ | Varchar(255) |  |

* DanhGia

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Giải thích | Kiểu dữ liệu | Khóa |
| 1 | id | Mã đánh giá | Int(11) | Khóa chính |
| 2 | id\_nguoi\_dung | người dùng đã đánh giá | Int(11) | Khóa ngoại |
| 3 | id\_bai\_do\_xe | bãi đỗ xe mà đánh giá áp dụng | Int(11) | Khóa ngoại |
| 4 | diem\_danh\_gia | điểm đánh giá của người dùng | Int(11) |  |
| 5 | noi\_dung | nội dung của đánh giá | Varchar(255) |  |
| 6 | thoi\_gian | thời điểm đánh giá được tạo | Datetime |  |

* GiaoDich

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Giải thích | Kiểu dữ liệu | Khóa |
| 1 | id | Mã giao dịch | Int(11) | Khóa chính |
| 2 | id\_nguoi\_dung | người dùng tham gia giao dịch | Int(11) | Khóa ngoại |
| 3 | id\_bai\_do\_xe | bãi đỗ xe được đặt | Int(11) | Khóa ngoại |
| 4 | thoi\_gian\_bat\_dau | thời gian bắt đầu | Datetime |  |
| 5 | thoi\_gian\_ket\_thuc | thời gian kết thúc | Datetime |  |
| 6 | gia | giá tiền của giao dịch | Float |  |
| 7 | trang\_thai | trạng thái của giao dịch | Varchar(255) |  |

* DangKyXe

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Giải thích | Kiểu dữ liệu | Khóa |
| 1 | id | Mã xe đăng ký | Int(11) | Khóa chính |
| 2 | id\_nguoi\_dung | người dùng đăng ký | Int(11) | Khóa ngoại |
| 4 | bien\_so | lưu trữ biển số xe | Varchar(255) |  |

* TinhThanh

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Giải thích | Kiểu dữ liệu | Khóa |
| 1 | id | Mã tỉnh thành | Int(11) | Khóa chính |
| 2 | ten | Tên tỉnh thành | Varchar(255) | Khóa ngoại |
| 4 | latitude | giá trị tọa độ vĩ độ | DECIMAL(10, 8) |  |
| 5 | longitude | giá trị tọa độ kinh độ | DECIMAL(11, 8) |  |

### Một số giao diện hệ thống

* Trang đăng ký tài khoản

A picture containing text, screenshot, logo, brand

Description automatically generated

Hình 2‑4: Trang đăng ký của hệ thống

Trang đăng ký tài khoản bãi đỗ xe được thiết kế để cung cấp cho người sử dụng một trải nghiệm nhanh chóng và dễ dàng. Bằng cách tạo một tài khoản, người sử dụng sẽ có thể tận hưởng nhiều ưu đãi và tiện ích mà ứng dụng mang đến.

* Trang đăng nhập người dùng

A screenshot of a parking login

Description automatically generated with medium confidence

Hình 2‑5: Trang đăng nhập của hệ thống

Để đăng nhập vào hệ thống người dung chỉ cần điền thông tin bao gồm email hoặc tên người dùng và mật khẩu. Trang được được thiết kế với gian diện đơn giản để người dùng có thể bắt đầu ngay bất kỳ nhiệm vụ nào người dung đang làm.

* Trang chủ

A screenshot of a map

Description automatically generated

Hình 2‑6: Màn hình chính của khách hàng khi sử dụng ứng dụng

Trang chủ là nơi người dùng có thể dễ dàng tìm kiếm và đặt chỗ bãi đỗ xe theo ngày, đồng thời khám phá các vị trí bãi đỗ xe trong khu vực mong muốn. Trang chủ của được thiết kế với một bản đồ tương tác, giúp người dùng có cái nhìn tổng quan về các vị trí bãi đỗ xe. Bên cạnh đó có thể di chuyển trên bản đồ, phóng to hoặc thu nhỏ để xem chi tiết khu vực mà bạn quan tâm. Người dùng cũng có thể sử dụng công cụ tìm kiếm để nhập địa điểm cụ thể hoặc địa điểm gần đó. Bên cạnh đó một chức năng được đưa vào đó là đặt chỗ dựa trên ngày và thời gian. Người dùng có thể chọn ngày và thời gian mong muốn và nhận danh sách các bãi đỗ xe có sẵn trong khoảng thời gian đó. Thông tin chi tiết về mỗi bãi đỗ xe bao gồm địa chỉ, giá cả, loại hình dịch vụ và tiện ích đi kèm giúp người dùng có lựa chọn phù hợp với nhu cầu của mình. Có thể thấy tiềm năng phát triển với việc mở rộng và cải thiện các tính năng hiện có. Nhóm đang nghiên cứu và phát triển các tính năng để hoàn thiện hơn.

# KẾT QUẢ NGHIÊN CỨU

### Các công cụ, công nghệ sử dụng để xây dựng ứng dụng

Công cụ: Dùng IDE intellij để code và biên dịch, postman để test api,xampp mysql để lưu trữ dữ liệu.

Công nghệ: Dùng javascript boot, spring core, spring data jpa để xử lý backend HTML, CSS, JS để xử lý giao diện.

*Java* là một ngôn ngữ lập trình phổ biến và mạnh mẽ được sử dụng rộng rãi trong phát triển ứng dụng. Nó có cú pháp dễ đọc, hỗ trợ đa nền tảng và có một cộng đồng lớn với nhiều tài liệu và công cụ hỗ trợ.

*Spring Boot* là một framework phát triển ứng dụng Java, xây dựng trên nền tảng Spring. Nó giúp đơn giản hóa quy trình phát triển bằng cách cung cấp các cấu hình mặc định và tự động cấu hình cho ứng dụng của bạn, giảm thiểu các công đoạn cấu hình và giúp bạn tập trung vào việc viết mã chức năng.

Có một số lý do mà người ta thích sử dụng Java và Spring Boot:

* Độ tin cậy và khả năng mở rộng: Java được sử dụng trong các hệ thống ứng dụng lớn, có khả năng xử lý lượng dữ liệu lớn và hỗ trợ khả năng mở rộng.
* Cộng đồng hỗ trợ: Java và Spring Boot có một cộng đồng lớn với nhiều tài liệu, diễn đàn và công cụ hỗ trợ, giúp bạn giải quyết các vấn đề và tìm kiếm thông tin.
* Tiêu chuẩn công nghiệp: Java được sử dụng rộng rãi trong nhiều lĩnh vực và có nhiều chuẩn công nghiệp. Spring Boot là một framework phát triển ứng dụng được chứng minh và sử dụng phổ biến trong cộng đồng phát triển.
* Tích hợp dễ dàng: Spring Boot cung cấp cơ chế tiêm phụ thuộc (dependency injection) mạnh mẽ, giúp bạn quản lý các thành phần ứng dụng và tích hợp dễ dàng với các công nghệ khác như cơ sở dữ liệu, giao diện người dùng, bộ nhớ đệm, v.v.

Tuy nhiên, việc chọn công nghệ phù hợp phụ thuộc vào yêu cầu cụ thể của dự án và sự quen thuộc của bạn với công nghệ đó. Nếu bạn đã quen thuộc với Java và muốn xây dựng các ứng dụng đáng tin cậy và linh hoạt, thì Java và Spring Boot là sự lựa chọn tốt.

Spring Data JPA là một phần mở rộng của Spring Framework giúp đơn giản hóa việc truy vấn và tương tác với cơ sở dữ liệu trong ứng dụng Java. Nó cung cấp một cách tiếp cận tự động hoá và tiện lợi cho việc truy vấn dữ liệu thông qua các phương thức CRUD (Create, Read, Update, Delete) và các truy vấn tùy chỉnh.

Các chức năng và lợi ích của Spring Data JPA:

* Tự động cấu hình: Spring Data JPA giúp tự động cấu hình và tạo các truy vấn SQL dựa trên các phương thức định nghĩa trong repository interface. Điều này giảm thiểu công việc lập trình viên phải viết và duy trì các truy vấn SQL thủ công.
* Tích hợp dễ dàng: Spring Data JPA tích hợp tốt với các công cụ và framework khác của Spring như Spring Boot, Spring MVC và Spring Security. Điều này giúp việc phát triển ứng dụng dễ dàng và nhanh chóng.
* Hỗ trợ đa cơ sở dữ liệu: Spring Data JPA không chỉ hỗ trợ cấu hình và tương tác với cơ sở dữ liệu quan hệ truyền thống như MySQL, PostgreSQL, Oracle, mà còn hỗ trợ các cơ sở dữ liệu NoSQL như MongoDB và Redis.
* Tối ưu hiệu suất: Spring Data JPA cung cấp các cơ chế tối ưu hiệu suất như bộ nhớ đệm (caching) và truy vấn lười biếng (lazy loading) để giảm thiểu tải cho cơ sở dữ liệu và cải thiện hiệu suất ứng dụng.

Tuy nhiên, cũng có một số phần hại tiềm năng của Spring Data JPA:

* Phụ thuộc vào ORM: Spring Data JPA phụ thuộc vào các công cụ ORM (Object-Relational Mapping) như Hibernate để tương tác với cơ sở dữ liệu. Điều này có thể làm tăng độ phức tạp và khó khăn khi gặp vấn đề phức tạp hoặc tùy chỉnh đặc biệt.
* Hiệu suất không tối ưu: Mặc dù Spring Data JPA cung cấp các cơ chế tối ưu hiệu suất, nhưng trong một số trường hợp đặc biệt, việc sử dụng các truy vấn tùy chỉnh phức tạp hoặc lớp mô hình đối tượng phức tạp có thể ảnh hưởng đến hiệu suất.
* Spring Security là một framework mạnh mẽ và phổ biến trong việc xác thực, ủy quyền và bảo mật ứng dụng Java. Nó cung cấp các tính năng và chức năng để bảo vệ ứng dụng khỏi các cuộc tấn công bảo mật và quản lý quyền truy cập của người dùng.

Chức năng chính của Spring Security:

* Xác thực (Authentication): Spring Security hỗ trợ xác thực người dùng thông qua nhiều cách khác nhau như đăng nhập form, xác thực OAuth, xác thực dựa trên token, v.v. Nó đảm bảo rằng người dùng là ai họ cho là, thông qua việc xác minh thông tin đăng nhập và xác thực giấy phép.
* Ủy quyền (Authorization): Spring Security cung cấp cơ chế ủy quyền để quản lý quyền truy cập của người dùng trong ứng dụng. Bằng cách định nghĩa các quy tắc và vai trò, bạn có thể kiểm soát và giới hạn quyền truy cập vào các tài nguyên và chức năng cụ thể.
* Bảo vệ chống tấn công: Spring Security cung cấp các tính năng bảo vệ chống lại các cuộc tấn công bảo mật như tấn công bằng cross-site scripting (XSS), tấn công bằng cross-site request forgery (CSRF), tấn công brute force, v.v. Nó cung cấp các cơ chế để kiểm tra và ngăn chặn các cuộc tấn công này.

Các lợi ích của Spring Security:

* Bảo mật toàn diện: Spring Security cung cấp các tính năng và cơ chế bảo mật mạnh mẽ để bảo vệ ứng dụng khỏi các cuộc tấn công và xâm nhập.
* Dễ tích hợp: Spring Security tích hợp tốt với các công nghệ và framework khác trong hệ sinh thái Spring như Spring Boot và Spring MVC. Việc tích hợp và cấu hình Spring Security trong ứng dụng Spring là khá dễ dàng.
* Quản lý quyền truy cập linh hoạt: Spring Security cho phép bạn định nghĩa và quản lý quyền truy cập của người dùng thông qua các vai trò và quy tắc. Điều này giúp kiểm soát chính xác và linh hoạt với quyền truy cập vào các tài nguyên trong ứng dụng.

Tuy nhiên, cũng có một số phần hại tiềm năng của Spring Security:

* Độ phức tạp: Do tính năng và linh hoạt của nó, việc cấu hình và sử dụng Spring Security có thể phức tạp đối với những người mới bắt đầu. Cần hiểu rõ về cơ chế và quy tắc bảo mật của nó để triển khai đúng và hiệu quả.
* Tiềm ẩn lỗ hổng bảo mật: Một cấu hình không chính xác hoặc một lỗi lập trình trong ứng dụng có thể dẫn đến các lỗ hổng bảo mật. Điều này đặc biệt quan trọng khi triển khai các cấu hình tùy chỉnh hoặc tích hợp với các công nghệ khác.

Tóm lại, Spring Security cung cấp các tính năng và chức năng bảo mật mạnh mẽ cho ứng dụng Java, giúp bảo vệ ứng dụng khỏi các cuộc tấn công và quản lý quyền truy cập của người dùng. Tuy nhiên, việc sử dụng và cấu hình nó cần được thực hiện cẩn thận để tránh các vấn đề bảo mật tiềm ẩn.

Mô hình 3 lớp (3-layer architecture) là một kiến trúc phân lớp phổ biến trong phát triển phần mềm. Nó chia ứng dụng thành ba lớp chính: Presentation Layer (giao diện người dùng), Business Logic Layer (logic kinh doanh) và Data Access Layer (truy cập dữ liệu). Dưới đây là lý do sử dụng, tác dụng và lợi ích của mô hình 3 lớp:

Lợi ích của Presentation Layer (giao diện người dùng):

* Tách biệt giao diện người dùng và logic kinh doanh: Presentation Layer chịu trách nhiệm hiển thị thông tin và tương tác với người dùng. Bằng cách tách biệt lớp này, bạn có thể dễ dàng thay đổi giao diện người dùng mà không ảnh hưởng đến logic kinh doanh bên trong.
* Tính linh hoạt và tùy chỉnh: Mô hình 3 lớp cho phép sử dụng nhiều công nghệ khác nhau để triển khai giao diện người dùng. Bạn có thể sử dụng các công nghệ web như HTML/CSS/JavaScript hoặc công nghệ desktop như Java Swing hoặc JavaFX.

Lợi ích của Business Logic Layer (logic kinh doanh):

* Tách biệt logic kinh doanh: Business Logic Layer chứa các quy tắc và quy trình logic kinh doanh của ứng dụng. Việc tách biệt lớp này giúp đảm bảo tính logic và tính nhất quán của ứng dụng.
* Tái sử dụng mã: Với Business Logic Layer riêng biệt, bạn có thể sử dụng lại các phần của nó trong các ứng dụng khác. Điều này giảm thiểu công sức và thời gian phát triển, đồng thời tăng tính nhất quán và độ tin cậy.

Lợi ích của Data Access Layer (truy cập dữ liệu):

* Tách biệt truy cập dữ liệu: Data Access Layer đảm nhận trách nhiệm truy cập và quản lý dữ liệu từ các nguồn dữ liệu như cơ sở dữ liệu, tệp tin, API, v.v. Tách biệt lớp này cho phép bạn thay đổi hoặc thay đổi nguồn dữ liệu mà không cần thay đổi logic kinh doanh.
* Tối ưu hiệu suất: Với Data Access Layer riêng biệt, bạn có thể tối ưu hóa các phương pháp truy cập dữ liệu, sử dụng bộ nhớ đệm (caching), hoặc triển khai các kỹ thuật tối ưu hóa truy vấn dữ liệu. Điều này giúp cải thiện hiệu suất ứng dụng.

Tóm lại, mô hình 3 lớp cung cấp sự tách biệt rõ ràng giữa giao diện người dùng, logic kinh doanh và truy cập dữ liệu.

Trong 40 năm qua, các nghiên cứu về mô hình lựa chọn bãi đỗ xe tập trung vào 3 hướng chính[6]:

Hướng đầu tiên dựa trên lý thuyết kinh tế để xây dựng bãi đỗ xe, quản lý phí và giảm tắc đường.

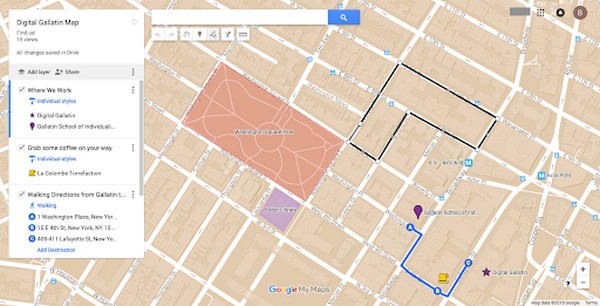
Hướng thứ hai là nghiên cứu quá trình lựa chọn bãi đỗ xe dựa trên hình thức đỗ xe (trên đường hoặc bãi đỗ), vị trí bãi đỗ và quá trình tìm kiếm bãi đỗ. Hành trình đi đến bãi đỗ được quyết định dựa trên hệ thống đường giao thông, khoảng cách và thời gian di chuyển, số lần chuyển hướng và số nút giao thông.

Hướng thứ ba tập trung vào sự liên hệ giữa việc lựa chọn bãi đỗ và tình hình giao thông, phân tích ảnh hưởng của quá trình tìm kiếm bãi đỗ đến lưu lượng trên tuyến. Các tiêu chí lựa chọn bãi đỗ xe trong đô thị được xem xét khá đầy đủ bao gồm thời gian

Để thực hiện được bài toán tìm hành trình tối ưu, quá trình tìm kiếm các phương án di chuyển được chia thành 2 bước: (1) tìm phương án tối ưu di chuyển bằng ô tô cho hành trình từ điểm xuất phát đến bãi đỗ; từ bãi đỗ đi điểm cuối hành trình và (2) tìm phương án đi bộ tối ưu từ bãi đỗ đến điểm cần đến và ngược lại.

Đối với bài toán tìm đường đi ngắn nhất trong lý thuyết đồ thị, có thể sử dụng thuật toán Dijkstra hoặc thuật toán Floyd-Warshall để giải.

Dijkstra là thuật toán sử dụng nhằm giải quyết bài toán đường đi ngắn nhất một nguồn (single source), đồ thị trọng số không âm[7-9]. Thuật toán Dijkstra cũng tối giản đường đi bằng cách xét các cạnh và so sánh 2 đường đi sẵn có với đường qua cả 3 đỉnh. Nguyên lý hoạt động bằng cách duy trì một tập hợp các đỉnh trong đó đã được biết chắc đường đi ngắn nhất. Qua từng bước, thuật toán sẽ chọn ra một đỉnh mà chắc chắn đã được tối ưu hóa cao nhất. Sau N bước, tất cả các đỉnh đều được chọn và mọi đường đi tìm được đều sẽ là ngắn nhất. Các ứng dụng tìm kiếm đường đi và chỉ đường hiện nay đều sẽ hiện nhiều lựa chọn với các trị số thời gian để bạn lựa chọn ra con đường ngắn nhất từ điểm xuất phát đến điểm đến dựa trên những hiển thị và các yếu tố tác động từ vệ tinh, từ đó áp dụng thuật toán Dijkstra để hiển thị đường.



Hình 3‑1 Ứng dụng Google Map với thuật toán Dijkstra

Thuật toán Floyd-Warshall sẽ giúp chúng ta giải quyết vấn đề hoạt động chưa hiệu quả khi phải chạy lặp đi lặp lại khá nhiều thao tác của một số thuật toán tìm được đi ngắn nhất. Floyd hoạt động được trên đồ thị có hướng, có thể có trọng số âm, tuy nhiên không có chu trình âm. Ngoài ra, Floyd còn có thể được dùng để phát hiện chu trình âm[10].

Sau khi chọn được bãi đỗ tối ưu theo hai mục tiêu chi phí và thời gian. Ứng dụng có thể xuất các thông tin: bãi đỗ, thông tin chi tiết  của bãi đỗ: số vị trí trống, giá thành đỗ,…

Với sự hỗ trợ của các công nghệ nhận dạng, cảm biến có thể tự động quá quản lý bãi đỗ xe. Các bãi đỗ xe thông minh (Smart Parking System) có thể cung cấp thông tin bãi đỗ, kết hợp với hệ thống thanh toán, quản lý đặt chỗ trực tuyến, công tác quản lý hoạt động đỗ xe được quản lý theo thời gian thực.

### Chương trình và kết quả

* Cấu hình bảo mật trong việc đăng nhập của hệ thống:



Hình 3‑2: Cài đặt cấu hình bảo vệ đăng nhập

Trên đây là một cấu hình bảo mật trong ứng dụng web sử dụng Spring Security. Bằng cách sử dụng @EnableWebSecurity và mở rộng từ lớp WebSecurityConfigurerAdapter, lớp SecurityConfig cho phép chúng ta cấu hình các tùy chọn bảo mật theo ý muốn. Phương thức configure(HttpSecurity http) được ghi đè để thiết lập chi tiết về bảo mật. Trong ví dụ này, nó bắt đầu bằng việc bật xác thực HTTP Basic bằng cách sử dụng .httpBasic(). Tiếp theo, chúng ta sử dụng .authorizeRequests() để khai báo các quy tắc truy cập. Trong đó, đường dẫn carparkingservice/parkinglot/create được giới hạn truy cập chỉ cho người dùng có vai trò "ADMIN" và phương thức POST bằng cách sử dụng .antMatchers(HttpMethod.POST,"carparkingservice/parkinglot/create").hasRole("ADMIN"). Phương thức configure(AuthenticationManagerBuilder auth) được ghi đè để cấu hình xác thực người dùng. Trong ví dụ này, chúng ta sử dụng xác thực trong bộ nhớ (in-memory authentication) với hai người dùng được định nghĩa: "user" với mật khẩu là "password" và vai trò "USER", cùng với "admin" có mật khẩu "password" và vai trò "ADMIN".

Tóm lại, đoạn code trên cho phép chúng ta cấu hình bảo mật trong ứng dụng web, sử dụng xác thực HTTP Basic và giới hạn quyền truy cập đối với một đường dẫn cụ thể.

* Cài đặt xử lý các yêu cầu liên quan đến "Admin" trong hệ thống

AdminController là một RESTful controller trong ứng dụng web. Nó xử lý các yêu cầu liên quan đến "Admin" trong hệ thống. @RestController đánh dấu AdminController là một controller RESTful, xử lý các yêu cầu HTTP và trả về dữ liệu dưới dạng JSON. @RequestMapping("carparkingservice/admin") định nghĩa tiền tố đường dẫn cho các yêu cầu tới AdminController là "carparkingservice/admin".@Autowired private AdminServiceImpl adminService tiêm AdminServiceImpl vào AdminController, cho phép truy cập các phương thức đã triển khai trong AdminServiceImpl. @RequestMapping(value = "create", method = RequestMethod.POST) xử lý yêu cầu POST tới "carparkingservice/admin/create". Nhận dữ liệu JSON và gọi phương thức create trong adminService để lưu trữ và trả về kết quả. @GetMapping("/read") xử lý yêu cầu GET tới "carparkingservice/admin/read". Nhận dữ liệu JSON và gọi phương thức read trong adminService để tìm và trả về đối tượng Admin dựa trên ID. @PostMapping("/update") xử lý yêu cầu POST tới "carparkingservice/admin/update". Nhận dữ liệu JSON và gọi phương thức update trong adminService để cập nhật đối tượng Admin và trả về kết quả. @DeleteMapping("{adminId}") xử lý yêu cầu DELETE tới "carparkingservice/admin/{adminId}". Nhận ID của đối tượng Admin và gọi phương thức delete trong adminService để xóa đối tượng và trả về thông báo kết quả. @GetMapping("/getAll") xử lý yêu cầu GET tới "carparkingservice/admin/getAll". Gọi phương thức findAll trong adminService để lấy danh sách tất cả đối tượng Admin và trả về.



Hình 3‑3: Cài đặt xử lý"Admin" trong hệ thống

* Cài đặt xử lý các yêu cầu liên quan đến "User" trong hệ thống



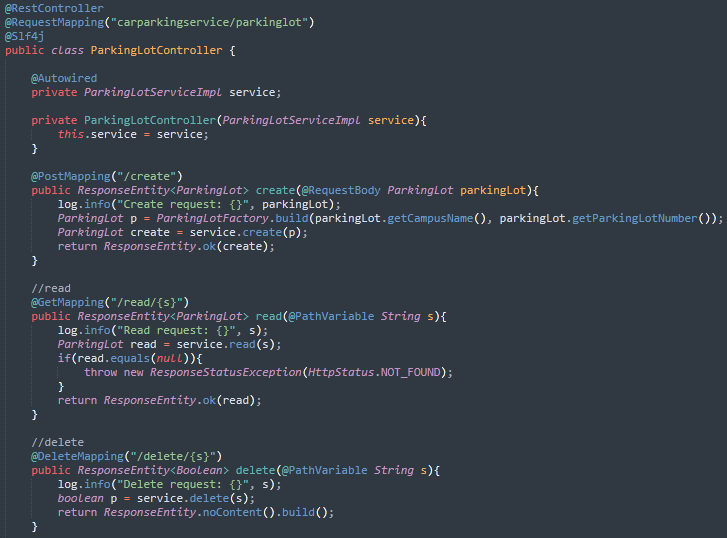
Hình 3‑4: Cài đặt xử lí User trong hệ thống

Cài đặt định nghĩa một RESTful controller UserController để xử lý các yêu cầu liên quan đến User trong hệ thống. Phương thức khởi tạo UserController được đánh dấu là private và nhận một đối tượng UserServiceImpl. Điều này đảm bảo rằng chỉ có một đối tượng UserServiceImpl được sử dụng trong UserController.

Phương thức create xử lý yêu cầu POST tới "/create". Nó nhận dữ liệu JSON thông qua @RequestBody và tạo một đối tượng User mới bằng cách sử dụng UserFactory. Sau đó, nó gọi phương thức create trong service để lưu trữ User và trả về kết quả dưới dạng ResponseEntity<User>.

Phương thức read xử lý yêu cầu GET tới "/read/{s}". Đối số @PathVariable {s} trong đường dẫn cho phép truyền giá trị vào phương thức. Nó gọi phương thức read trong service để tìm kiếm User dựa trên giá trị s. Nếu không tìm thấy User, ném một ngoại lệ ResponseStatusException với mã trạng thái HTTP "NOT\_FOUND". Nếu tìm thấy, nó trả về User đã tìm thấy bằng cách sử dụng ResponseEntity.ok().

* Cài đặt xử lí các yêu cầu liên quan đến bãi đỗ xe



Hình 3‑5:Cài đặt xử lí yêu cầu liên quan đến bãi đỗ xe

Cài đặt xử lí này gồm các phương thức:

create: Xử lý yêu cầu POST "/create" để tạo một ParkingLot mới. Nó nhận dữ liệu JSON và gọi phương thức create trong ParkingLotServiceImpl để lưu trữ và trả về kết quả.

read: Xử lý yêu cầu GET "/read/{s}" để đọc thông tin về ParkingLot. Giá trị {s} trong đường dẫn được truyền vào phương thức và sử dụng để tìm kiếm ParkingLot. Nếu không tìm thấy, ném một ngoại lệ ResponseStatusException. Nếu tìm thấy, trả về ParkingLot đã tìm thấy.

update: Xử lý yêu cầu POST "/update" để cập nhật thông tin của ParkingLot. Nó nhận dữ liệu JSON và gọi phương thức update trong ParkingLotServiceImpl để cập nhật và trả về kết quả.

delete: Xử lý yêu cầu DELETE "/delete/{s}" để xóa ParkingLot. Giá trị {s} trong đường dẫn được truyền vào phương thức và sử dụng để xóa ParkingLot. Trả về ResponseEntity.noContent() sau khi xóa thành công.

getAll: Xử lý yêu cầu GET "/getAll" để lấy danh sách tất cả ParkingLot. Gọi phương thức findAll trong ParkingLotServiceImpl và trả về danh sách ParkingLot.

Trên đây là một số cài đặt chính của hệ thống, về cơ bản thể thống đã được xây dựng để xử lí các vấn đề chức năng đã đề ra như: theo dõi, xử lý việc đặt bãi đỗ, quá trình quá trình đặt bãi đỗ, xem và tra cứu thông tin đơn đặt, em, theo dõi bãi đỗ, đơn đặt và khách hang, dễ hiểu, dễ dùng, hấp dẫn, dễ tìm kiếm thông tin bãi đỗ, đảm bảo an toàn tuyệt đối thông tin của khách hang. Tuy nhiên do thời gian còn hạn chế, nhóm nghiên cứu mới chỉ thực hiện được trên hình thức nhận và trả dữ liệu yêu cầu của người dùng và chưa hoàn thiện thành sản phẩm thực tế.

# KẾT LUẬN

Kết luận:

Nhóm đã dành đa số thời gian qua để thực hiện một cuộc nghiên cứu chi tiết và sâu sắc về hệ điều hành Android Automotive. Nghiên cứu này đã mở ra cánh cửa để có thể khám phá một thế giới mới, nơi mà công nghệ và ô tô hòa quyện thành một tương lai hứa hẹn.

Mục tiêu chính của việc nghiên cứu này là hiểu rõ về kiến trúc phức tạp và cấu trúc tinh vi của hệ điều hành Android Automotive. Nhóm nghiên cứu đã tiến hành nghiên cứu đối tượng, tìm hiểu về cách mà hệ điều hành này được thiết kế và triển khai trên các thiết bị ô tô. Việc khám phá và phân tích các thành phần cấu thành hệ điều hành đã giúp nhóm có cái nhìn toàn diện và sâu sắc hơn về cách thức hoạt động của nó.

Trong quá trình nghiên cứu, nhóm đã tìm hiểu và đánh giá các tính năng chính của hệ điều hành Android Automotive. Từ khả năng kết nối với các thiết bị, đến tích hợp các ứng dụng giải trí và thông tin giữa ô tô và điện thoại thông minh, chúng tôi đã thấy rằng Android Automotive thực sự mang đến một trải nghiệm tuyệt vời và tích cực cho người dùng.

Đặc biệt là khả năng tùy chỉnh và mở rộng của hệ điều hành này. Android Automotive cho phép các nhà phát triển xây dựng các ứng dụng và tích hợp chúng vào hệ thống ô tô một cách linh hoạt. Điều này tạo điều kiện thuận lợi cho việc cung cấp những trải nghiệm độc đáo và cá nhân hóa cho người dùng. Chúng tôi đã tìm hiểu cách để tận dụng tối đa khả năng tùy chỉnh này và đưa ra những đề xuất về cách phát triển ứng dụng và giao diện người dùng phù hợp với ngành công nghiệp ô tô.Android Automotive mang lại nhiều lợi ích đáng kể cho ngành công nghiệp ô tô. Một trong những lợi ích quan trọng nhất là khả năng tăng cường tích hợp và sự kết nối giữa các thành phần trong ô tô. Hệ điều hành này cung cấp một nền tảng linh hoạt để các nhà sản xuất ô tô và nhà phát triển ứng dụng có thể tương tác và làm việc cùng nhau để mang đến những trải nghiệm mới và cải tiến đột phá.Ngoài ra, có thể nhận thấy tiềm năng và ứng dụng rộng rãi của nghiên cứu này. Kết quả và kiến thức thu được có thể được áp dụng trong nhiều lĩnh vực khác nhau, từ việc phát triển các ứng dụng giải trí và thông tin trong ô tô đến tối ưu hóa hệ thống và tăng cường an ninh và bảo mật. Nhóm nghiên cứu tin rằng nghiên cứu này sẽ đóng góp tích cực vào sự tiến bộ của ngành công nghiệp ô tô và mang lại những trải nghiệm tốt hơn cho người dùng.

Việc xây dựng một hệ thống tìm kiếm bãi đỗ xe trên hệ điều hành Android Automotive hoàn chỉnh cần đi kèm với rất nhiều yếu tố. Tuy nhiên nếu được hoàn thiện và ứng dụng thực tế, ứng dụng này sẽ giúp người dùng tìm kiếm và đặt chỗ đỗ xe gần nhất một cách nhanh chóng, giúp tránh tình trạng mất thời gian và nguy cơ bị phạt khi đỗ xe không đúng quy định, giúp chủ các bãi đỗ xe quản lý tốt hơn và cung cấp thông tin chính xác đến người dùng về vị trí đỗ xe. Ứng dụng này không chỉ có lợi ích về mặt cá nhân mà còn mang lại nhiều lợi ích cho xã hội và kinh tế.

Hạn chế:

Mặc dù nhóm nghiên cứu đã tiến hành một cuộc nghiên cứu chi tiết và sâu sắc về hệ điều hành Android Automotive, chúng tôi cũng gặp phải một số hạn chế và khó khăn trong quá trình nghiên cứu.

Đầu tiên, nhóm nghiên cứu đã đối mặt với khó khăn trong việc xin dữ liệu do chưa có sự tin tưởng từ các chủ bãi xe. Lý do không tin tưởng này có thể bắt nguồn từ các vấn đề bảo mật hoặc khả năng chia sẻ thông tin. Điều này đã gây khó khăn trong việc thu thập dữ liệu và tạo ra một rào cản đối với quá trình nghiên cứu của nhóm.

Bên cạnh đó,Công nghệ và kiến thức yêu cầu để xây dựng ứng dụng trên nền tảng Android Automotive là mới mẻ và phức tạp. Điều này đòi hỏi nhóm nghiên cứu phải có kiến thức chuyên sâu và kinh nghiệm phong phú trong lĩnh vực này. Điều này có thể tạo ra khó khăn cho nhóm trong việc tiến hành đề tài nghiên cứu.

Tổng quát lại, mặc dù nhóm nghiên cứu đã đạt được những kết quả tích cực, nhưng cũng đã gặp một số hạn chế. Tuy nhiên, những khó khăn này sẽ tiếp tục là cơ hội để nhóm đánh giá và nâng cao năng lực trong tương lai, đồng thời đóng góp vào sự phát triển của lĩnh vực Android Automotive.

Kiến nghị và hướng phát triển đề tài:

Mô hình tìm kiếm bãi đỗ xe cần có sự đồng bộ dữ liệu từ hệ thống quản lý giao thông, hệ thống quản lý bãi đỗ xe thông minh, hệ thống dẫn hướng, cung cấp thông tin đỗ xe, hệ thống đặt chỗ và thanh toán tự động, các dữ liệu về đặc điểm ra quyết định chọn bãi đỗ v.v. Một vấn đề cần giải quyết là tạo được niềm tin từ các chủ bãi đỗ xe để họ sẵn sàng chia sẻ dữ liệu về các bãi đỗ trong thời gian thực, vì vậy vẫn cần có nhiều cải tiến hơn để đưa ra được ưu điểm vượt trội của đề tài.

# TÀI LIỆU THAM KHẢO

[1]. K.Chinetha, J.Daphney Joann, A.Shalini. February 2015. An Evolution of Android Operating System and Its Version.

[2]. TechCrunch, 22/09/2018, Google hợp tác với liên minh xe hơi lớn nhất thế giới, dự định đem Android lên hàng triệu chiếc xe ô tô

[3]. Developers Android. Last updated 2022-10-11. *Android Devices, Android Automotive OS documents, What is Android Automotive?*

[4]. James Dolan. June 17, 2022. *Android Auto vs. Android Automotive: What's the Difference?*

[5]. Peter Gessler (Android Automotive Architect), Tino Müller (Mobility Solutions), Marius Mailat (CTO). September 2, 2020. *Android Automotive OS White Paper.*

[6]. Dương Minh Châu. 02/10/2018. *Mô hình tìm kiếm bãi đỗ xe tối ưu sử dụng hệ thống đặt chỗ.*

[7]. aptech.fpt.edu.vn/. 16/1/2021. *Ứng dụng thực tiễn của thuật toán Dijkstra trong đời sống hiện nay.*

[8]. Heeket Mehta, Pratik Kanani, Priya Lande.  8, May 2019. *Google Maps*.

[9]. programiz.com/dsa/dijkstra-algorithm. January 22, 2019. *Dijkstra’s Algorithm pseudocode.*

[10]. Ramadiani , D Bukhori , Azainil and N Dengen. 2018. *Floyd-warshall algorithm to determine the shortest path based on android.*