**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**



**---🙠🕮🙢---**

**BÁO CÁO ĐỒ ÁN**

**THỰC TẬP CHUYÊN MÔN**

**XÂY DỰNG ỨNG DỤNG ĐẶT XE THÔNG MINH**

**Sinh viên thực hiện: Phạm Thị Ly - 6151071069**

**Lớp : CQ.K61.CNTT**

**Bộ môn Công nghệ thông tin**

**Người hướng dẫn:**

**Tp. Hồ Chí Minh, tháng 12 năm 2023**

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**



**---🙠🕮🙢---**

**BÁO CÁO ĐỒ ÁN**

**THỰC TẬP CHUYÊN MÔN**

**XÂY DỰNG ỨNG DỤNG ĐẶT XE THÔNG MINH**

**Sinh viên thực hiện: Phạm Thị Ly Nam, Nữ: Nữ Dân tộc: Kinh**

**Lớp: CQ.K61.CNTT Ngành học: Công nghệ thông tin**

**Năm thứ: 4/4**

**Người hướng dẫn:**

**Tp. Hồ Chí Minh, tháng 12 năm 2023**

**LỜI NHẬN XÉT**

**..................................................................................................................................................**

**..................................................................................................................................................**

**..................................................................................................................................................**

**..................................................................................................................................................**

**..................................................................................................................................................**

**..................................................................................................................................................**

**..................................................................................................................................................**

**..................................................................................................................................................**

**..................................................................................................................................................**

**..................................................................................................................................................**

**..................................................................................................................................................**

**..................................................................................................................................................**

**..................................................................................................................................................**

**..................................................................................................................................................**

**..................................................................................................................................................**

**..................................................................................................................................................**

**..................................................................................................................................................**

**..................................................................................................................................................**

**..................................................................................................................................................**

**..................................................................................................................................................**

**..................................................................................................................................................**

**..................................................................................................................................................**

**..................................................................................................................................................**

**..................................................................................................................................................**

**..................................................................................................................................................**

*Tp. Hồ Chí Minh, ngày…tháng…năm…*

Giảng viên hướng dẫn

**MỤC LỤC**

[**MỤC LỤC** **4**](#_Toc86847799)

[**DANH MỤC HÌNH ẢNH** **6**](#_Toc86847799)

[**DANH MỤC BẢNG BIỂU** **.**](#_Toc86847799)

[**LỜI MỞ ĐẦU** **.**](#_Toc86847799)

[**CHƯƠNG 1: TỔNG QUAN** **.**](#_Toc86847800)

[1.1 Giới thiệu chung về đề tài .](#_Toc86847801)

[1.2 Tình hình nghiên cứu. .](#_Toc86847802)

[1.3 Mục tiêu .](#_Toc86847803)

[1.4 Phương pháp nghiên cứu .](#_Toc86847804)

[1.5 Phạm vi của đề tài .](#_Toc86847805)

[1.6 Ý nghĩa của đề tài .](#_Toc86847806)

[**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT** **.**](#_Toc86847808)

[**2.1 Tổng quan về ngôn ngữ lập trình Dart** **.**](#_Toc86847809)

[*2.1.1 Dart là gì ?* **.**](#_Toc86847810)

[*2.1.2 Cú pháp cơ bản của Dart* **.**](#_Toc86847810)

[*2.1.3 Kiểu dữ liệu trong Dart* **.**](#_Toc86847810)

[*2.1.4 Cấu trúc điều khiển và vòng lặp trong Dart* **.**](#_Toc86847810)

[*2.1.5 Hàm và lớp trong Dart* **.**](#_Toc86847810)

[*2.1.6 Ưu điểm của Dart.* **.**](#_Toc86847811)

[*2.1.7 Lý do nên chọn ngôn ngữ lập trình Dart.* **.**](#_Toc86847812)

[**2.2 Phần mềm Android Studio.** **.**](#_Toc86847813)

[*2.2.1 Android Studio và phát triển ứng dụng di động* **.**](#_Toc86847814)

[*2.2.2 Cách cấu hình Android Studio.* **.**](#_Toc86847815)

[*2.2.3 Giao diện và các thành phần chính .* **.**](#_Toc86847816)

[*2.2.4 Tạo dự án Flutter trong Android Studio.* **.**](#_Toc86847816)

[*2.2.5 Cách cấu hình thiết bị ảo.* **.**](#_Toc86847816)

[**2.3 Framework Flutter** **.**](#_Toc86847817)

[*2.3.1 Khái niệm về Flutter và ưu điểm.* **.**](#_Toc86847818)

[*2.3.2 Cài đặt Flutter và thiết lập môi trường phát triển.* **.**](#_Toc86847819)

[*2.3.3 Cấu trúc dự án Flutter và cách tổ chức mã nguồn.* **.**](#_Toc86847819)

[*2.3.4 Widget trong Flutter và cách sử dụng chúng.* **.**](#_Toc86847819)

[**2.4 Flutter map và cách tương tác với bản đồ** **.**](#_Toc86847817)

[*2.4.1 Giới thiệu Flutter map và các tính năng.* **.**](#_Toc86847818)

[*2.4.2 Cách hiển thị bản đồ trong ứng dụng Flutter.* **.**](#_Toc86847819)

[*2.4.3 Tạo điểm đánh dấu và đa điểm trên bản đồ.* **.**](#_Toc86847819)

[*2.4.4 Xử lí sự kiện tương tác với bản đồ trong Flutter*](#_Toc86847819)

[**2.5 Firebase và tính năng của Firebase**](#_Toc86847817)

[*2.5.1 Firebase và lợi ích trong ứng dụng Flutter*](#_Toc86847818)

[*2.5.2 Cấu hình Firebase trong dự án Flutter..* **.**](#_Toc86847819)

[*2.5.3 Tạo và quản lý dữ liệu Firebase Realtime Database* **.**](#_Toc86847819)

[*2.5.4 Firebase Authentication và xác thực người dùng* **.**](#_Toc86847819)

[*2.5.5 Các tính năng khác của Firebase* **.**](#_Toc86847819)

[**CHƯƠNG 3: PHÂN TÍCH ĐỀ TÀI VÀ ỨNG DỤNG ĐẶT XE** **.**](#_Toc86847820)

[**3.1 Đặc tả bài toán** **.**](#_Toc86847821)

[*3.1.1 Tổng quan về bài toán đặt xe.* **.**](#_Toc86847819)

[*3.1.2 Sơ đồ mô tả hệ thống.* **.**](#_Toc86847819)

[*3.1.3 Xác định yêu cầu chức năng và phi chức năng của hệ thống*](#_Toc86847823)

[*3.1.4 Định nghĩa, khái niệm và thuật ngữ quan trong trong lĩnh vực đặt xe*](#_Toc86847824)

[**3.2 Phát triển ứng dụng đặt xe thông minh**](#_Toc86847832)

[*3.2.1 Quy trình phát triển, thiết kế giao diện ứng dụng* **.**](#_Toc86847833)

[*3.2.2 Triển khai và kiểm thử* **.**](#_Toc86847834)

[**CHƯƠNG 4: KẾT LUẬN VÀ KIẾN NGHỊ** **.**](#_Toc86847865)

[4.1 Kết quả đạt được: **.**](#_Toc86847866)

[4.2 Hạn chế tồn tại: **.**](#_Toc86847867)

[4.3 Hướng phát triển: **.**](#_Toc86847868)

[4.4 Kết luận: **.**](#_Toc86847869)

[Tài liệu tham khảo: **.**](#_Toc86847870)

**DANH MỤC HÌNH ẢNH**

[Hình 2.1.1: Ngôn ngữ lập trình Dart 17](#_Toc86847799)

[Hình 2.2.3: Giao diện chính Android Studio 21](#_Toc86847799)

[Hình 2.2.5: Thiết bị Android ảo 21](#_Toc86847799)

[Hình 2.4.2 1: Thêm phiên bản Flutter Map 26](#_Toc86847799)

[Hình 2.4.2 2: Thư viện import 26](#_Toc86847799)

[Hình 2.4.2 3: Cấu hình Flutter map 26](#_Toc86847799)

[Hình 3.1.2: Sơ đồ mô tả hệ thống 5](#_Toc86847799)

[Hình 1.1 5](#_Toc86847799)

[Hình 1.1 5](#_Toc86847799)

[Hình 1.1 5](#_Toc86847799)

[Hình 1.1 5](#_Toc86847799)

[Hình 1.1 5](#_Toc86847799)

[Hình 1.1 5](#_Toc86847799)

[Hình 1.1 5](#_Toc86847799)

[Hình 1.1 5](#_Toc86847799)

[Hình 1.1 5](#_Toc86847799)

[Hình 1.1 5](#_Toc86847799)

**DANH MỤC BẢNG BIỂU**

[Hình 1.1 5](#_Toc86847799)

[Hình 1.1 5](#_Toc86847799)

[Hình 1.1 5](#_Toc86847799)

[Hình 1.1 5](#_Toc86847799)

[Hình 1.1 5](#_Toc86847799)

[Hình 1.1 5](#_Toc86847799)

[Hình 1.1 5](#_Toc86847799)

**LỜI MỞ ĐẦU**

Trên khắp thế giới, nhu cầu di chuyển đã trở thành một phần không thể thiếu trong cuộc sống hiện đại. Đặc biệt, trong các đô thị đông đúc, việc tìm kiếm phương tiện giao thông thuận tiện và nhanh chóng đã trở thành một thách thức đối với nhiều người. Tuy nhiên, với sự phát triển của công nghệ di động, người dùng ngày càng có nhiều cơ hội tiếp cận các ứng dụng đặt xe di động, giúp giải quyết vấn đề di chuyển một cách hiệu quả và tiện lợi.

Trong bối cảnh xã hội ngày càng phát triển, việc sử dụng ứng dụng đặt xe di động đã trở thành một xu hướng không thể phủ nhận. Thay vì phải gọi điện thoại hoặc chờ đợi tìm kiếm một phương tiện giao thông, người dùng có thể dễ dàng truy cập vào ứng dụng trên điện thoại di động và chọn xe theo ý muốn của mình. Từ Taxi, xe ôm, đến xe tự lái, các ứng dụng đặt xe di động cung cấp một loạt các lựa chọn phù hợp với nhu cầu của mọi người.

Dẫn chứng cho sự phổ biến của ứng dụng đặt xe di động là sự thành công của các công ty như Uber, Grab, Be, Gojek, Lyft,.... Những hãng công nghệ này đã tạo ra một cách tiếp cận hoàn toàn mới cho việc di chuyển, đồng thời mang lại lợi ích lớn cho cả người dùng và tài xế. Người dùng có thể dễ dàng tìm kiếm và đặt xe chỉ trong vài phút, đồng thời nhận được thông tin về tài xế, giá cả và thời gian đến. Đối với tài xế, ứng dụng đặt xe di động giúp tăng thu nhập và tối ưu hóa việc điều phối công việc.

Sự cấp thiết của đề tài này rõ ràng. Ứng dụng đặt xe di động không chỉ đơn giản là một công nghệ mới, mà còn mang lại nhiều lợi ích cho cộng đồng. Việc sử dụng ứng dụng đặt xe giúp giảm thiểu sự tắc nghẽn giao thông, giảm lượng khí thải và tiết kiệm thời gian cho người dùng. Ngoài ra, nó cũng mở ra cơ hội kinh doanh cho tài xế và góp phần thúc đẩy nền kinh tế địa phương.

Với những lợi ích đáng kể mà ứng dụng đặt xe di động mang lại, không ngạc nhiên khi ngày càng có nhiều doanh nghiệp và cá nhân tìm cách phát triển các ứng dụng tương tự. Tuy nhiên, việc thành công trong lĩnh vực này không chỉ đòi hỏi sự đổi mới công nghệ, mà còn yêu cầu cung cấp trải nghiệm người dùng tốt và đảm bảo an ninh thông tin.

Qua việc nghiên cứu và thảo luận về đề tài này, tôi hy vọng sẽ có cái nhìn sâu hơn về vai trò và ảnh hưởng của ứng dụng đặt xe di động trong cuộc sống hiện đại. Hãy cùng bắt đầu khám phá thế giới của ứng dụng đặt xe di động và tìm hiểu cách nó đã và đang thay đổi cách chúng ta di chuyển và tương tác với thành phố xung quanh.

Trong tài liệu này, chúng ta sẽ khám phá các khía cạnh quan trọng và công cụ cần thiết để phát triển một ứng dụng đặt xe di động hiệu quả.

* Chương 1 giới thiệu về đề tài, bao gồm tình hình nghiên cứu, mục tiêu, phạm vi và ý nghĩa của đề tài.
* Chương 2 tập trung vào cơ sở lý thuyết, giới thiệu về ngôn ngữ lập trình Dart, phần mềm Android Studio, framework Flutter, Flutter map và tích hợp Firebase vào ứng dụng.
* Chương 3 tiến xa hơn với phân tích đề tài và ứng dụng, đặc tả bài toán đặt xe, hướng giải quyết, quy trình phát triển ứng dụng và các khía cạnh quan trọng trong việc phát triển ứng dụng đặt xe.
* Cuối cùng, chương 4 trình bày kết luận và kiến nghị dựa trên kết quả đạt được từ nghiên cứu, cũng như đề cập đến hạn chế tồn tại và hướng phát triển tiếp theo của ứng dụng.

Tài liệu tham khảo được cung cấp để người đọc có thể tìm hiểu thêm về các nguồn thông tin và tài liệu liên quan đến đề tài.

Báo cáo này là kết quả của quá trình nghiên cứu và làm đồ án thực tập chuyên môn của tôi, dựa trên lý thuyết được học trên lớp và nỗ lực tìm hiểu sâu hơn ngoài thực tế. Tôi xin cảm ơn sự hướng dẫn và hỗ trợ tận tình từ Thầy Trần Phong Nhã, Cô Trần Thị Dung, Thầy Trần Quốc Khánh và các giảng viên bộ môn công nghệ thông tin.

Báo cáo này là sản phẩm độc lập của em và phản ánh quan điểm cá nhân. Tuy nhiên, không thể tránh khỏi những sai sót không cố ý. Tôi rất mong nhận được những phản hồi tích cực từ quý thầy cô để tôi có thể cải thiện.

# **CHƯƠNG 1: TỔNG QUAN**

## **1.1 Giới thiệu chung đề tài**

Trong thời đại công nghệ hiện đại, ứng dụng di động đã trở thành một phần không thể thiếu trong cuộc sống hàng ngày của chúng ta. Chúng đã thay đổi cách chúng ta làm việc, giải trí và tương tác với thế giới xung quanh. Một trong những ứng dụng di động nổi bật và ngày càng phổ biến là ứng dụng đặt xe.

Ứng dụng đặt xe di động là một công nghệ đột phá trong lĩnh vực giao thông và di chuyển. Với sự tiện lợi và linh hoạt, chúng đã thay đổi cách chúng ta tìm kiếm và sử dụng các phương tiện giao thông. Thay vì phải chờ đợi lâu để tìm một chiếc taxi hoặc gọi điện thoại để đặt xe, chúng ta có thể dễ dàng truy cập vào ứng dụng trên điện thoại di động và chọn phương tiện theo ý muốn của mình. Điều này mang lại sự tiện lợi, linh hoạt và tiết kiệm thời gian cho người dùng.

Ứng dụng đặt xe di động không chỉ giúp người dùng tìm kiếm và đặt xe một cách dễ dàng, mà còn cung cấp thông tin chi tiết về tài xế, giá cả, thời gian đến và quãng đường di chuyển. Người dùng có thể xem đánh giá từ người khác và lựa chọn phương tiện phù hợp với nhu cầu của mình. Đồng thời, ứng dụng đặt xe cũng mang lại lợi ích cho tài xế, giúp tăng thu nhập và tối ưu hóa việc điều phối công việc.

Ngoài ra, ứng dụng đặt xe di động còn giúp giải quyết một số vấn đề trong giao thông đô thị. Chúng giúp giảm thiểu tình trạng ùn tắc giao thông bằng cách tối ưu hóa việc chia sẻ phương tiện và giảm lượng xe cá nhân trên đường. Điều này đồng nghĩa với việc giảm lượng khí thải và ô nhiễm môi trường, đồng thời tạo ra một môi trường giao thông sạch hơn và bền vững.

Trên khắp thế giới, ứng dụng đặt xe di động đã trở thành một lực lượng thay đổi cách chúng ta di chuyển và tương tác với thành phố. Với sự phát triển không ngừng, các ứng dụng đặt xe đã mở ra nhiều cơ hội kinh doanh và đem lại lợi ích cho cả người dùng và các nhà cung cấp dịch vụ giao thông.

Trong đồ án này, chúng ta sẽ khám phá sâu hơn về ứng dụng đặt xe di động, tìm hiểu về nguồn gốc, phát triển và những ảnh hưởng tích cực mà chúng mang lại. Chúng ta cũng sẽ xem xét các thách thức và cơ hội trong lĩnh vực này, đồng thời dự báo về tương lai phát triển của ứng dụng đặt xe.

**1.2 Tình hình nghiên cứu**

Khảo sát thực trạng đề cập đến nhu cầu di chuyển của người dùng hiện tại. Việc hiểu rõ về tình hình này là cực kỳ quan trọng để xác định các vấn đề, hạn chế và cơ hội liên quan đến ứng dụng đặt xe. Dưới đây là một số yếu tố quan trọng cần khảo sát:

* Nghiên cứu về ứng dụng đặt xe và công nghệ di động: Có nhiều nghiên cứu đã tập trung vào việc phân tích và đánh giá các ứng dụng đặt xe di động và công nghệ di động liên quan. Những nghiên cứu này thường tập trung vào các khía cạnh như tiện lợi, trải nghiệm người dùng, tác động đến giao thông đô thị và cách ứng dụng di động thay đổi cách mọi người tương tác với dịch vụ đặt xe.
* Hiệu suất và tối ưu hóa: Một lĩnh vực quan trọng trong nghiên cứu ứng dụng đặt xe là tối ưu hóa cách thức hoạt động của hệ thống. Các nhà nghiên cứu đã tìm hiểu về cách cải thiện hiệu suất của ứng dụng, bao gồm việc tối ưu hóa quy trình đặt xe, tối ưu hóa việc phân bổ phương tiện và tối ưu hóa định tuyến.
* Tác động đến giao thông và môi trường: Nghiên cứu về ứng dụng đặt xe cũng đã tập trung vào tác động của chúng đến giao thông đô thị và môi trường. Các nghiên cứu này thường xem xét tác động của ứng dụng đặt xe đến lưu lượng giao thông, tình trạng ùn tắc, khả năng chia sẻ phương tiện và giảm thiểu lượng khí thải.
* Đánh giá trải nghiệm người dùng: Một khía cạnh quan trọng của nghiên cứu về ứng dụng đặt xe là đánh giá trải nghiệm người dùng. Các nghiên cứu đã tìm hiểu về sự hài lòng của người dùng, yếu tố ảnh hưởng đến sự chọn lựa ứng dụng đặt xe, và cách cải thiện trải nghiệm người dùng thông qua giao diện người dùng và tính năng mới.
* Các thách thức và cơ hội: Nghiên cứu cũng đã tập trung vào các thách thức và cơ hội trong lĩnh vực ứng dụng đặt xe. Các thách thức bao gồm vấn đề về bảo mật và quyền riêng tư, cạnh tranh trong thị trường ứng dụng đặt xe, và quản lý tài xế và phương tiện. Cơ hội bao gồm khai thác dữ liệu để cung cấp dịch vụ tốt hơn, phát triển các mô hình kinh doanh mới và mở rộng ứng dụng đặt xe sang các lĩnh vực khác như giao hàng và chia sẻ phương tiện.

Tổng quan về tình hình nghiên cứu về đề tài ứng dụng đặt xe cho thấy rằng đây là một lĩnh vực đang trong quá trình phát triển nhanh chóng. Có nhiều cơ hội để nghiên cứu và cải tiến các khía cạnh khác như của ứng dụng đặt xe, từ trải nghiệm người dùng đến tối ưu hóa hệ thống và tác động lên giao thông và môi trường. Tuy nhiên, cũng có nhiều thách thức cần được giải quyết, bao gồm các vấn đề về bảo mật, quản lý tài xế và quyền riêng tư.

Trong tương lai, nghiên cứu về ứng dụng đặt xe có thể tập trung vào việc phân tích dữ liệu lớn và sử dụng trí tuệ nhân tạo để cung cấp dịch vụ đặt xe thông minh và tối ưu hơn. Các nghiên cứu có thể nghiên cứu về phát triển mô hình kinh doanh mới và ứng dụng đặt xe trong các lĩnh vực khác như giao hàng và chia sẻ phương tiện.

Tóm lại, tình hình nghiên cứu về ứng dụng đặt xe đã đạt được nhiều tiến bộ và đang tiếp tục phát triển. Có nhiều cơ hội để nghiên cứu và cải thiện các khía cạnh khác nhau của ứng dụng đặt xe, từ trải nghiệm người dùng đến hiệu suất hệ thống và tác động lên giao thông và môi trường.

## **1.3 Mục tiêu**

Xây dựng một ứng dụng đặt xe thông minh hiệu quả trong các vấn đề thiết yếu của người dùng. Các mục tiêu cụ thể của nghiên cứu bao gồm:

* Đánh giá trải nghiệm người dùng: Nghiên cứu nhằm đánh giá và cải thiện trải nghiệm người dùng trong việc sử dụng ứng dụng đặt xe. Mục tiêu có thể bao gồm nghiên cứu về yếu tố ảnh hưởng đến sự hài lòng của người dùng, khả năng tìm kiếm và đặt xe dễ dàng, tính minh bạch của thông tin về giá cả và tình trạng xe, và sự thuận tiện trong quá trình thanh toán.
* Tối ưu hóa quy trình đặt xe: Mục tiêu là tìm hiểu và tối ưu hóa quy trình đặt xe để giảm thiểu thời gian chờ đợi và tăng khả năng đáp ứng nhu cầu của người dùng. Nghiên cứu này có thể tập trung vào việc phân tích và cải thiện các bước từ tìm kiếm xe, đặt xe, xác nhận và nhận xe.
* Tối ưu hóa phân bổ phương tiện: Mục tiêu là nghiên cứu và phát triển các thuật toán tối ưu để phân bổ phương tiện một cách hiệu quả. Nghiên cứu có thể tập trung vào việc tối ưu hóa việc cân đối giữa cung cấp và nhu cầu, tối ưu hóa thời gian và khoảng cách di chuyển của phương tiện, và tối ưu hóa việc phân bổ phương tiện dựa trên các yếu tố như sự tiện ích và hiệu suất năng lượng.
* Tác động đến giao thông và môi trường: Mục tiêu là đánh giá tác động của ứng dụng đặt xe đến giao thông và môi trường. Nghiên cứu có thể tập trung vào việc phân tích tác động của ứng dụng đặt xe đến lưu lượng giao thông, tình trạng ùn tắc, khả năng giảm thiểu lượng xe trống chạy, và tác động đến ô nhiễm môi trường và khí thải.
* Quản lý tài xế và phương tiện: Mục tiêu là nghiên cứu và đề xuất các phương pháp quản lý tài xế và phương tiện hiệu quả. Nghiên cứu có thể tập trung vào việc phân tích và cải thiện quy trình đăng ký, đào tạo và đánh giá tài xế, cải thiện quy trình bảo trì và sửa chữa phương tiện, và phát triển các phương thức quản lý hiệu suất tài xế và phương tiện.
* Bảo mật và quyền riêng tư: Mục tiêu là nghiên cứu và đề xuất các giải pháp bảo mật và quyền riêng tư trong ứng dụng đặt xe. Nghiên cứu có thể tập trung vào việc phân tích và cải thiện bảo mật dữ liệu người dùng, quản lý quyền truy cập và xác thực, và đảm bảo sự riêng tư của thông tin cá nhân trong quy trình đặt xe.

## **1.4 Phương pháp nghiên cứu**

Nghiên cứu các tài liệu và các phương pháp nghiên cứu:

* Phân tích yêu cầu và thiết kế hệ thống.
* Phát triển ứng dụng và kiểm thử.
* Đánh giá trải nghiệm người dùng.
* Tối ưu hóa hiệu suất và độ tin cậy.
* Phân tích dữ liệu và học máy
* Nghiên cứu các trang web hướng dẫn lập trình.
* Trao đổi kiến thức từ giảng viên và học tập từ các anh, chị, bạn bè cùng lớp.

## **1.5 Phạm vi của đề tài**

Phần mềm xây dựng chủ yếu  là tập trung vào việc phát triển một ứng dụng di động đặt xe sử dụng nền tảng Flutter và tích hợp các dịch vụ của Firebase. Đề tài tập trung vào các khía cạnh như thiết kế giao diện người dùng, tích hợp hệ thống đặt xe, quản lý thông tin xe và người dùng, xử lý thanh toán, cung cấp thông tin vị trí và tích hợp các tính năng của Firebase như quản lý người dùng, cơ sở dữ liệu realtime, và phân tích hiệu suất ứng dụng.

## **1.6 Ý nghĩa của đề tài**

Ý nghĩa khoa học:

* Đóng góp kiến thức về phát triển ứng dụng di động: Đề tài này mang lại cơ hội nghiên cứu và ứng dụng các công nghệ mới như Flutter và Firebase trong việc phát triển ứng dụng di động. Việc tìm hiểu và áp dụng các nguyên lý và phương pháp phát triển ứng dụng trong bối cảnh cụ thể này có thể cung cấp thông tin và kinh nghiệm quý giá cho cộng đồng phát triển phần mềm di động.
* Nâng cao hiểu biết về tích hợp dịch vụ Cloud: Sự kết hợp giữa Flutter và Firebase trong đề tài này giúp nghiên cứu và hiểu rõ hơn về việc tích hợp và sử dụng dịch vụ đám mây trong phát triển ứng dụng di động. Điều này cung cấp kiến thức và kỹ năng liên quan đến cách tương tác với các dịch vụ như quản lý người dùng, cơ sở dữ liệu realtime và phân tích hiệu suất.
* Đưa ra các phương pháp phát triển và kiểm thử ứng dụng: Nghiên cứu này có thể đóng góp các phương pháp, kỹ thuật và quy trình phát triển và kiểm thử ứng dụng di động. Từ việc thiết kế giao diện người dùng cho đến quản lý dữ liệu và tích hợp dịch vụ, các phương pháp nghiên cứu trong đề tài có thể cung cấp hướng dẫn và khung làm việc cho những người muốn phát triển ứng dụng tương tự.

Ý nghĩa thực tiễn:

* Cải thiện trải nghiệm người dùng: Đề tài này tạo ra một ứng dụng đặt xe với Flutter và Firebase, có thể cung cấp một trải nghiệm người dùng tốt hơn trong việc đặt xe và quản lý thông tin liên quan. Việc phát triển một ứng dụng đặt xe tiện ích và dễ sử dụng có thể giúp cải thiện sự thuận tiện và trải nghiệm của người dùng khi đi lại.
* Ứng dụng thực tế trong ngành công nghiệp: Ứng dụng đặt xe là một lĩnh vực thực tế và có ảnh hưởng đáng kể trong ngành công nghiệp giao thông. Việc phát triển một ứng dụng đặt xe sử dụng công nghệ Flutter và tích hợp Firebase có thể mang lại lợi ích thực tế và giúp cải thiện quy trình đặt xe và quản lý trong ngành.
* Khả năng mở rộng và mở cửa cho phát triển tương lai: Việc nghiên cứu về việc sử dụng Flutter và Firebase trong ứng dụng di động mở ra cơ hội mở rộng và phát triển ứng dụng trong tương lai. Kiến thức và kinh nghiệm thu thập được từ đề tài này có thể áp dụng cho các dự ánứng dụng di động khác, không chỉ trong lĩnh vực đặt xe mà còn trong nhiều lĩnh vực ứng dụng khác sử dụng Flutter và Firebase.

**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

**2.1 Tổng quan về ngôn ngữ lập trình Dart.**

*2.1.1 Dart là gi?*



Hình 2.1.1 Ngôn ngữ lập trình Dart

Dart là một ngôn ngữ lập trình phát triển bởi Google, được thiết kế để xây dựng các ứng dụng web, ứng dụng di động và ứng dụng máy tính. Dart là một ngôn ngữ đa mô hình, có thể được sử dụng để phát triển các ứng dụng đơn luồng và đa luồng.

Một số đặc điểm của Dart bao gồm:

* Dart là một ngôn ngữ có cú pháp gần giống với các ngôn ngữ lập trình như C, C++ và Java, điều này giúp cho việc học và chuyển đổi giữa các ngôn ngữ dễ dàng hơn đối với những người đã có kinh nghiệm với các ngôn ngữ này.
* Dart hỗ trợ kiểu tĩnh và kiểu động. Điều này cho phép lập trình viên lựa chọn giữa việc xác định kiểu biến tại thời điểm biên dịch hoặc để ngôn ngữ loại bỏ kiểm tra kiểu động.
* Dart có hệ thống gom rác tự động, giúp quản lý bộ nhớ tự động và giảm công việc của lập trình viên trong việc quản lý bộ nhớ thủ công.
* Dart hỗ trợ việc xây dựng ứng dụng đa nền tảng thông qua Flutter, một framework phát triển ứng dụng di động và web sử dụng Dart. Flutter cho phép lập trình viên xây dựng giao diện người dùng đẹp mắt và tương tác trên nhiều nền tảng như Android, iOS, web và desktop.

Dart được sử dụng rộng rãi trong cộng đồng phát triển ứng dụng di động và web, đặc biệt là khi sử dụng cùng với Flutter để xây dựng các ứng dụng di động đa nền tảng. Dart cung cấp một cú pháp dễ hiểu, hiệu suất cao và khả năng phát triển linh hoạt, làm cho nó trở thành một lựa chọn hấp dẫn cho việc phát triển ứng dụng hiện đại.

*2.1.2 Cú pháp cơ bản của Dart.*

Các cú pháp cơ bản của ngôn ngữ Dart:

* Khai báo biến.
* Cú pháp điều khiển if-else.
* Cú pháp vòng lặp for.
* Cú pháp vòng lặp while.
* Cú pháp hàm
* Cú pháp khai báo lớp.

*2.1.3 Kiểu dữ liệu trong Dart.*

Các kiểu dữ liệu cơ bản trong Dart bao gồm:

* Số nguyên: int
* Số thực: double
* Chuỗi: String
* Boolean: bool
* Danh sách: List
* Bộ sưu tập: Set
* Bảng băm: Map

*2.1.4 Cấu trúc điều khiển và vòng lặp trong Dart*

* Cấu trúc điều khiển if-else: sử dụng để thực hiện các lệnh dựa trên một điều kiện.
* Vòng lặp for: sử dụng để lặp qua một tập hợp các phần tử cho trước.
* Vòng lặp while: sử dụng để lặp lại một khối lệnh trong khi một điều kiện cụ thể đúng.

*2.1.5 Hàm và lớp trong Dart*

* Hàm trong Dart được sử dụng để nhóm mã thành các đơn vị có thể tái sử dụng. Một hàm có thể nhận đối số và trả về một giá trị.
* Lớp trong Dart là một bản thiết kế cho đối tượng, bao gồm các thuộc tính và phương thức. Lớp được sử dụng để tạo các đối tượng và định nghĩa hành vi của chúng.

*2.1.6 Ưu điểm của Dart*

Dart có nhiều ưu điểm kể đến như:

* Dart hỗ trợ kiểu tĩnh và kiểu động, cho phép kiểm tra kiểu biên dịch và khả năng kiểm tra kiểu động tùy chọn.
* Dart có hệ thống gom rác tự động, giúp quản lý bộ nhớ tự động và giảm công việc quản lý bộ nhớ thủ công.
* Dart cung cấp cú pháp gần giống với các ngôn ngữ lập trình phổ biến như C, C++, và Java, giúp người lập trình dễ dàng học và chuyển đổi giữa các ngôn ngữ này.
* Dart hỗ trợ việc xây dựng ứng dụng đa nền tảng thông qua Flutter, cho phép phát triển ứng dụng di động cho cả iOS và Android.

*2.1.7 Lý do nên chọn ngôn ngữ lập trình Dart*

Dart là một ngôn ngữ lập trình đơn giản, hiệu suất cao và hỗ trợ đa nền tảng, đặc biệt phù hợp cho việc phát triển ứng dụng di động sử dụng Flutter.

* Đơn giản và dễ học: Dart có cú pháp gần giống với nhiều ngôn ngữ lập trình phổ biến, giúp người lập trình dễ dàng tiếp cận và học nhanh.
* Hiệu suất cao: Với việc được biên dịch trước thành mã máy, Dart cung cấp hiệu suất cao và tốc độ thực thi nhanh.
* Hỗ trợ đa nền tảng: Dart kết hợp với Flutter, một framework phát triển ứng dụng di động đa nền tảng, cho phép xây dựng ứng dụng di động cho cả iOS và Android từ một mã nguồn duy nhất.
* Mạnh mẽ và linh hoạt: Dart hỗ trợ tính năng mạnh mẽ như hệ thống kiểu tĩnh và động, async/await cho lập trình bất đồng bộ, và hỗ trợ đối tượng và lập trình hàm.
* Cộng đồng và tài liệu phong phú: Dart có một cộng đồng lớn và nhiều tài liệu hỗ trợ, giúp người lập trình dễ dàng tìm kiếm thông tin và hỗ trợ khi cần thiết.

**2.2 Phần mềm Android Studio.**

*2.2.1 Android Studio và phát triển ứng dụng di động.*

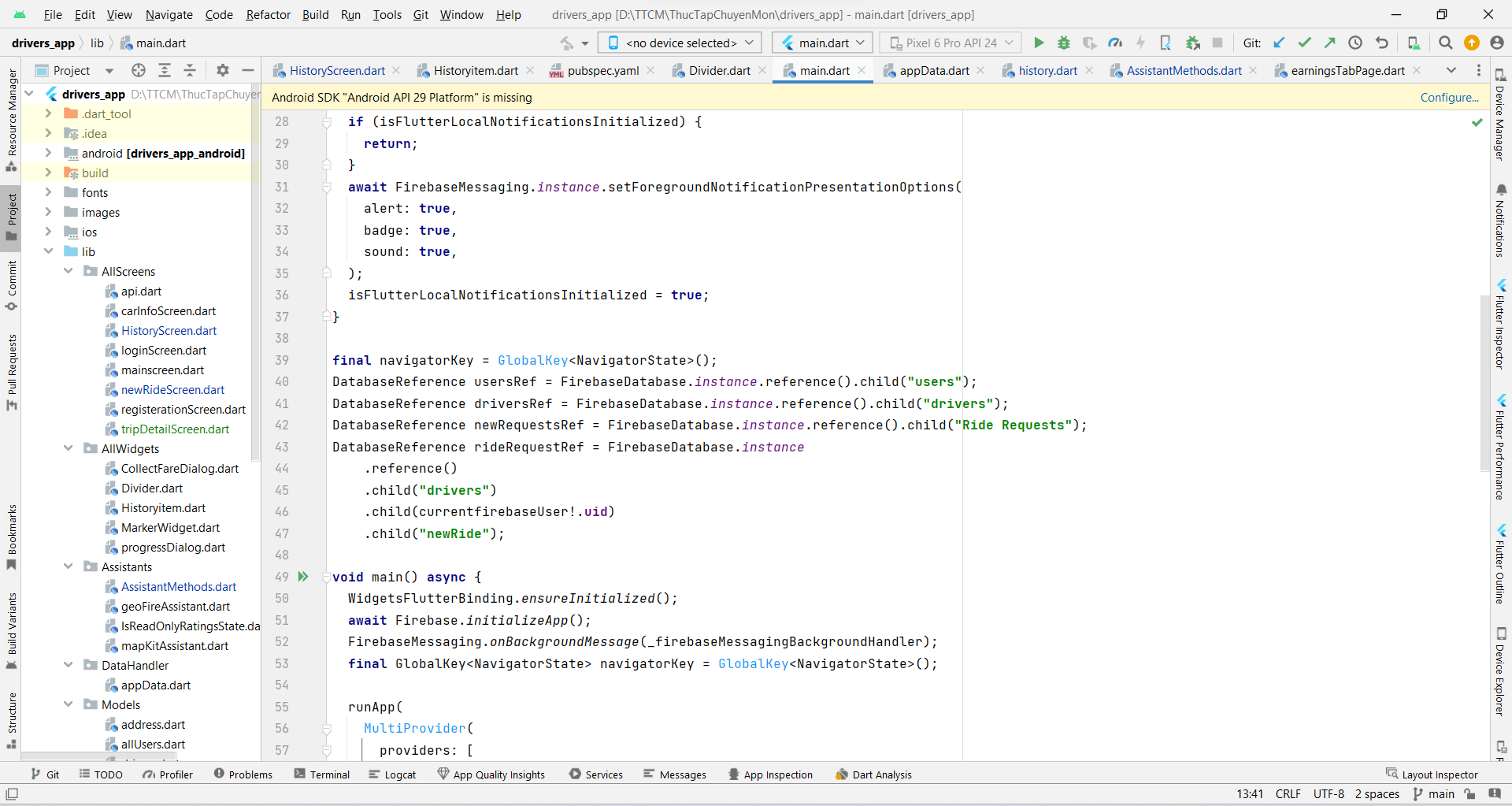
Android Studio là một môi trường phát triển tích hợp (IDE) chính thức dành cho việc phát triển ứng dụng di động Android. Nó cung cấp các công cụ và tài nguyên cần thiết để xây dựng, kiểm thử và triển khai ứng dụng di động trên nền tảng Android. Android Studio được xây dựng trên nền tảng IntelliJ IDEA và hỗ trợ ngôn ngữ Java và Kotlin cho việc lập trình ứng dụng Android.

*2.2.2 Cách cấu hình Android Studio.*

Để cấu hình Android Studio, bạn có thể làm theo các bước sau:

1. Tải xuống và cài đặt Android Studio từ trang web chính thức của Android Developer (<https://developer.android.com/studio>).
2. Sau khi cài đặt, khởi động Android Studio và làm theo hướng dẫn trên màn hình để hoàn thành quá trình cài đặt và cấu hình ban đầu.
3. Android Studio sẽ yêu cầu tải xuống và cài đặt một số thành phần bổ sung như SDK, bộ công cụ Gradle và các gói hỗ trợ. Bạn có thể lựa chọn các tùy chọn mà bạn muốn và tiếp tục quá trình cài đặt.
4. Sau khi quá trình cài đặt hoàn tất, Android Studio sẽ mở một cửa sổ chào đón và bạn sẽ sẵn sàng bắt đầu phát triển ứng dụng di động trên nền tảng Android.

*2.2.3 Giao diện và các thành phần chính.*



Hình 2.2.3 Giao diện chính Android Studio

*2.2.4 Tạo dự án Flutter trong Android Studio.*

Để tạo một dự án Flutter trong Android Studio, bạn có thể làm theo các bước sau:

1. Mở Android Studio và chọn "CreateNew Project" trên màn hình chào đón hoặc từ menu "File" -> "New" -> "New Project".
2. Trong hộp thoại "Create New Project", chọn "Flutter" trong danh sách các mẫu dự án.
3. Đặt tên cho dự án của bạn và chọn vị trí lưu trữ.
4. Chọn phiên bản Flutter và Dart mà bạn muốn sử dụng trong dự án.
5. Tiếp theo, bạn có thể chọn một số cài đặt khác nhau cho dự án của mình, chẳng hạn như kiến trúc ứng dụng và ngôn ngữ.
6. Sau khi hoàn thành các cài đặt, nhấn "Finish" để tạo dự án Flutter mới.
7. Android Studio sẽ tạo dự án Flutter và tự động tải các gói phụ thuộc cần thiết từ Dart Pub và Flutter Pub.

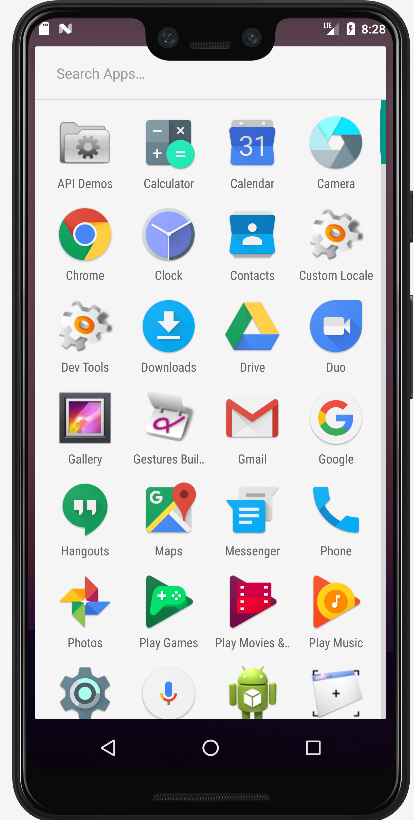
*2.2.5 Cách cấu hình thiết bị ảo.*

Để cấu hình thiết bị ảo trong Android Studio, bạn có thể làm theo các bước sau:

1. Mở Android Studio và chọn "AVD Manager" từ thanh công cụ hoặc từ menu "Tools" -> "AVD Manager".
2. Trong cửa sổ AVD Manager, chọn "Create Virtual Device".
3. Trên màn hình "Select Hardware", bạn có thể chọn một thiết bị ảo có sẵn từ danh sách, hoặc bạn có thể chọn "New Hardware Profile" để tạo một thiết bị ảo tùy chỉnh.
4. Sau khi chọn một thiết bị ảo, nhấn "Next" để chọn phiên bản hệ điều hành Android cho thiết bị ảo.
5. Tiếp theo, bạn có thể chọn hình ảnh hệ điều hành Android từ danh sách các hình ảnh có sẵn hoặc nhấp vào "Download" để tải xuống các hình ảnh khác từ Google Play.
6. Sau khi chọn hình ảnh hệ điều hành, nhấn "Next" và đặt tên cho thiết bị ảo của bạn.
7. Cuối cùng, nhấn "Finish" để tạo thiết bị ảo.

Trở lại cửa sổ AVD Manager, bạn sẽ thấy thiết bị ảo của mình hiển thị trong danh sách. Bạn có thể khởi động thiết bị ảo bằng cách nhấp vào nút "Play" trong cột "Actions".

Sau khi thiết bị ảo được khởi động, bạn có thể chạy ứng dụng của mình trên thiết bị ảo bằng cách chọn thiết bị trong cửa sổ "Run Configuration" và nhấn nút "Run" hoặc "Debug" trên thanh công cụ.



Hình 2.2.5 Thiết bị Android ảo

**2.3 Framework Flutter.**

*2.3.1 Khái niệm về Flutter và ưu điểm.*

Flutter là một framework mã nguồn mở phát triển bởi Google, được sử dụng để xây dựng giao diện người dùng đẹp và tương tác trên nền tảng di động, web và máy tính. Với Flutter, bạn chỉ cần viết một lần mã nguồn và triển khai nó trên nhiều nền tảng khác nhau mà không cần phải viết lại mã nguồn cho từng nền tảng riêng biệt.

Ưu điểm của Flutter bao gồm:

* Hiệu suất cao: Flutter sử dụng nguyên bản các thành phần giao diện người dùng, giúp ứng dụng chạy mượt mà và nhanh chóng.
* Giao diện người dùng đẹp: Flutter cung cấp một bộ công cụ phong phú cho thiết kế giao diện người dùng, cho phép bạn tạo ra giao diện đẹp và tùy chỉnh theo ý muốn.
* Phát triển nhanh: Flutter hỗ trợ tái sử dụng mã nguồn, cho phép bạn phát triển và triển khai ứng dụng nhanh chóng.
* Hot Reload: Flutter cung cấp tính năng Hot Reload, cho phép bạn xem kết quả ngay lập tức khi thay đổi mã nguồn, giúp tăng tốc quá trình phát triển.
* Hỗ trợ đa nền tảng: Flutter cho phép bạn triển khai ứng dụng trên Android, iOS, web và desktop từ cùng một mã nguồn.

### *2.3.2 Cài đặt Flutter và thiết lập mô trường phát triển.*

Để cài đặt Flutter và thiết lập môi trường phát triển, bạn có thể làm theo các bước sau:

1. Tải xuống Flutter từ trang web chính thức của Flutter ([https://flutter.dev](https://flutter.dev/)).
2. Giải nén tệp tải xuống vào vị trí bạn muốn cài đặt Flutter.
3. Đặt biến môi trường FLUTTER\_HOME để trỏ đến thư mục chứa Flutter.
4. Thêm đường dẫn đến thư mục Flutter vào biến môi trường PATH.
5. Mở terminal và chạy lệnh flutter doctor để kiểm tra cài đặt và cấu hình của Flutter.
6. Flutter doctor sẽ hiển thị thông báo về các thành phần cần thiết và các vấn đề cài đặt. Bạn cần giải quyết các vấn đề này trước khi tiếp tục.
7. Để phát triển ứng dụng Flutter, bạn cần cài đặt một trình biên dịch Dart. Flutter doctor sẽ cung cấp liên kết để tải xuống trình biên dịch Dart nếu nó chưa được cài đặt.

*2.3.3 Cấu trúc dự án Flutter và cách tổ chức mã nguồn.*

Dự án Flutter có cấu trúc tổ chức mã nguồn sau:

* Thư mục android: Chứa mã nguồn và tệp cấu hình liên quan đến phần Android của ứng dụng Flutter.
* Thư mục ios: Chứa mã nguồn và tệp cấu hình liên quan đến phần iOS của ứng dụng Flutter.
* Thư mục lib: Chứa mã nguồn Dart cho giao diện người dùng và logic của ứng dụng.
* Tệp pubspec.yaml: Đây là tệp cấu hình cho dự án Flutter, trong đóđịnh nghĩa các phụ thuộc và tài nguyên được sử dụng trong dự án.
* Thư mục test: Chứa các tệp tin để viết và chạy các bài kiểm tra đơn vị và tích hợp cho ứng dụng.
* Khi tổ chức mã nguồn trong dự án Flutter, bạn có thể tạo các thư mục con bổ sung trong thư mục lib để phân chia mã nguồn thành các phần khác nhau, như các mô-đun, thành phần giao diện người dùng, chức năng, v.v. Điều này giúp tổ chức mã nguồn một cách rõ ràng và dễ quản lý.

*2.3.4 Widget trong Flutter và cách sử dụng.*

Trong Flutter, mọi thứ đều được coi là một widget, từ các thành phần giao diện như nút (button), hộp (container), văn bản (text) đến các thành phần phức tạp hơn như danh sách cuộn (scrollable list), hộp chuyển đổi (switch), v.v. Mỗi widget đều có một giao diện (interface) và có thể chứa các widget con bên trong nó.

Flutter cung cấp một loạt các widget được xây dựng sẵn để ta có thể sử dụng, và cũng có thể tạo widget tùy chỉnh của riêng mình. Một số widget phổ biến trong Flutter bao gồm:

MaterialApp: Widget này cung cấp một ứng dụng Material Design, bao gồm các thành phần như thanh ứng dụng, đường dẫn (navigation), vv.

* Scaffold: Widget này cung cấp một giao diện chung cho ứng dụng, bao gồm thanh tiêu đề, thanh ứng dụng, nội dung chính và các thành phần khác.
* Container: Widget này đại diện cho một hộp trên giao diện người dùng và có thể chứa nội dung hoặc các widget con khác.
* Text: Widget này hiển thị một đoạn văn bản trên giao diện người dùng.
* Image: Widget này hiển thị một hình ảnh trên giao diện người dùng.
* ListView: Widget này tạo ra một danh sách cuộn chứa các widget con.
* RaisedButton và FlatButton: Đây là các widget nút cho phép tương tác với người dùng.
* TextField: Widget này cho phép người dùng nhập dữ liệu vào một trường văn bản.
* Checkbox và Switch: Đây là các widget cho phép người dùng chọn một hoặc nhiều tùy chọn.
* AppBar: Widget này tạo ra một thanh tiêu đề trên giao diện người dùng.

Để sử dụng một widget trong Flutter, bạn có thể tạo một đối tượng của nó và thêm nó vào cây widget của ứng dụng. Bạn có thể tạo các widget con bên trong các widget khác để xây dựng giao diện người dùng phức tạp hơn.

**2.4 Flutter Map và cách tương tác với bản đồ.**

*2.4.1 Giới thiệu Flutter Map và các tính năng.*

Flutter Map là một thư viện mã nguồn mở được sử dụng để hiển thị và tương tác với bản đồ trong ứng dụng Flutter. Nó cung cấp các công cụ và API để hiển thị các loại bản đồ khác nhau, bao gồm bản đồ raster (ảnh), bản đồ vectơ và bản đồ tùy chỉnh. Flutter Map hỗ trợ đa nền tảng và cho phép bạn tương tác với bản đồ bằng cách thêm điểm đánh dấu, vẽ đa giác, xử lý sự kiện bản đồ và nhiều tính năng khác.

Các tính năng của Flutter Map bao gồm:

* Hiển thị bản đồ: Flutter Map hỗ trợ hiển thị các loại bản đồ khác nhau, bao gồm bản đồ Google, bản đồ OpenStreetMap, bản đồ Mapbox và nhiều loại bản đồ tùy chỉnh khác.
* Điểm đánh dấu và đa điểm: Bạn có thể thêm điểm đánh dấu và đa điểm trên bản đồ để hiển thị thông tin địa lý và tương tác với chúng.
* Vẽ các hình dạng đa giác: Flutter Map cho phép bạn vẽ các đa giác trên bản đồ để biểu diễn các khu vực, vùng đất hoặc tùy chỉnh các khu vực.
* Xử lý sự kiện bản đồ: Bạn có thể xử lý các sự kiện tương tác với bản đồ như nhấp chuột, kéo và di chuyển để thực hiện các hành động tương ứng trong ứng dụng của mình.

*2.4.2 Cách hiển thị bản đồ trong ứng dụng Flutter.*

Để hiển thị bản đồ trong ứng dụng Flutter sử dụng Flutter Map, bạn cần thực hiện các bước sau:

1. Thêm thư viện Flutter Map vào tệp pubspec.yaml của dự án:



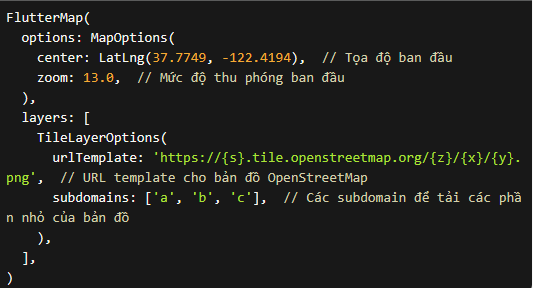
Hình 2.4.2 1 Thêm phiên bản Flutter Map

1. Chạy lệnh "flutter packages get" để cập nhật các phụ thuộc.
2. Trong mã nguồn Dart của ứng dụng, import thư viện Flutter Map:



Hình 2.4.2 2 Thư viện import

1. Tạo Widget Map trong cây widget của ứng dụng và cấu hình nó với các thông số như loại bản đồ, tọa độ ban đầu và thuộc tính khác:



Hình 2.4.2 3 Cấu hình Flutter Map

Trong ví dụ trên, chúng ta sử dụng bản đồ OpenStreetMap và tọa độ ban đầu là San Francisco. Đặt Widget Map trong một Container hoặc Scaffold để hiển thị trên giao diện người dùng.

*2.4.3 Tạo điểm đánh dấu và đa điểm trên bản đồ.*

Để tạo điểm đánh dấu và đa điểmtrên bản đồ sử dụng Flutter Map, bạn có thể làm như sau:

* Tạo một danh sách các điểm đánh dấu hoặc đa điểm mà bạn muốn hiển thị trên bản đồ. Mỗi điểm đánh dấu hoặc đa điểm được đại diện bởi một đối tượng Marker hoặc Polygon trong Flutter Map.
* Trong Widget Map, một MarkerLayerOptions hoặc PolygonLayerOptions được thêm vào danh sách các layers để hiển thị các điểm đánh dấu hoặc đa điểm trên bản đồ.

*2.4.4 Xử lí sự kiện tương tác với bản đồ trong Flutter.*

Để xử lý sự kiện tương tác với bản đồ trong Flutter Map, bạn có thể sử dụng các hàm xử lý sự kiện được cung cấp bởi MapOptions trong Widget Map. Ngoài ra onTap, MapOptions cũng cung cấp các hàm xử lý sự kiện khác như onLongPress, onDoubleTap, onMove,....Bạn có thể sự dụng các hàm này để xử lý các sự kiện tương tác khác trên bản đồ.

**2.5 Firebase và tính năng của Firebase.**

*2.5.1 Firebase và lợi ích trong ứng dụng Flutter.*

Firebase là một nền tảng phát triển ứng dụng di động và web do Google cung cấp. Nó cung cấp một loạt các dịch vụ đám mây để giúp phát triển ứng dụng chất lượng cao, bao gồm cơ sở dữ liệu realtime, xác thực người dùng, lưu trữ, thông báo đẩy và nhiều tính năng khác. Trong ứng dụng Flutter, Firebase mang lại nhiều lợi ích quan trọng như:

* Phát triển nhanh chóng: Firebase cung cấp các dịch vụ đã được xây dựng sẵn và tích hợp tương thích tốt với Flutter, giúp bạn phát triển ứng dụng nhanh chóng mà không cần xây dựng hạ tầng phức tạp từ đầu.
* Tính ổn định và mở rộng: Firebase được xây dựng trên cơ sở hạ tầng đám mây của Google, đảm bảo tính ổn định và khả năng mở rộng cao cho ứng dụng của bạn, ngay cả khi có hàng triệu người dùng sử dụng.
* Tích hợp sẵn với công cụ phát triển: Firebase tích hợp tốt với các công cụ phát triển phổ biến như Android Studio và Visual Studio Code, giúp bạn dễ dàng phát triển và thử nghiệm ứng dụng.
* Tài liệu phong phú: Firebase cung cấp tài liệu chi tiết, ví dụ mã và hướng dẫn sử dụng, giúp bạn nhanh chóng làm quen và triển khai các tính năng của Firebase trong ứng dụng Flutter của mình.

*2.5.2 Cấu hình Firebase trong duẹ án Flutter.*

Để sử dụng Firebase trong dự án Flutter, bạn cần thực hiện các bước sau:

1. Đăng nhập hoặc tạo tài khoản Firebase tại <https://firebase.google.com/>.
2. Tạo một dự án Firebase mới và đặt tên cho dự án.
3. Sau khi tạo dự án, bạn sẽ nhận được tệp cấu hình Firebase (google-services.json cho Android hoặc GoogleService-Info.plist cho iOS) từ trang cấu hình dự án Firebase của bạn.
4. Trong dự án Flutter, thêm tệp cấu hình Firebase vào thư mục chính của dự án (thường là android/app cho Android và ios/Runner cho iOS).
5. Chỉnh sửa tệp pubspec.yaml của dự án Flutter và thêm các phụ thuộc Firebase cần thiết.

*2.5.3 Tạo và quản lý dữ liệu Firebase Realtime Database.*

Firebase Realtime Database là một cơ sở dữ liệu thời gian thực được cung cấp bởi Firebase. Để tạo và quản lý dữ liệu trong Firebase Realtime Database, bạn có thể sử dụng SDK Firebase cho Flutter.

*2.5.4 Firebase Authentication và xác thực người dùng.*

Firebase Authentication cung cấp phương thức xác thực người dùng cho ứng dụng của bạn. Bạn có thể sử dụng Firebase Authentication trong ứng dụng Flutter của mình để cho phép người dùng đăng ký, đăng nhập và đăng xuất bằng nhiều phương thức khác nhau như email và mật khẩu, số điện thoại, tài khoản Google, Facebook, Twitter, và nhiều hơn nữa. Firebase Authentication cung cấp một cách dễ dàng để quản lý thông tin người dùng, xác thực và phân quyền truy cập vào các tính năng của ứng dụng.

Khi sử dụng Firebase Authentication trong ứng dụng Flutter, bạn cần thực hiện các bước sau:

1. Tạo dự án Firebase và kết nối ứng dụng Flutter của bạn với dự án Firebase đó. Đảm bảo rằng bạn đã tạo và cấu hình dịch vụ Firebase Authentication trong dự án của mình.
2. Cài đặt gói firebase\_auth trong file pubspec.yaml của dự án Flutter
3. Import package firebase\_auth vào file Dart của ứng dụng Flutter
4. Sử dụng các phương thức của FirebaseAuth để thực hiện các tác vụ xác thực người dùng.
5. Tương tự, để đăng nhập bằng email và mật khẩu, bạn có thể sử dụng phương thức signInWithEmailAndPassword

*2.5.5 Các tính năng khác của Firebase.*

Ngoài Firebase Realtime Database và Firebase Authentication, Firebase còn cung cấp rất nhiều tính năng khác hữu ích cho ứng dụng Flutter của bạn. Dưới đây là một số tính năng quan trọng khác của Firebase:

* Firebase Cloud Firestore: Là một cơ sở dữ liệu đám mây linh hoạt và mạnh mẽ, cung cấp khả năng lưu trữ và đồng bộ dữ liệu dễ dàng và phức tạp hơn Firebase Realtime Database.
* Firebase Cloud Messaging (FCM): Cho phép gửi thông báo đẩy tới người dùng trên các thiết bị di động và web. Bạn có thể gửi thông báo đến tất cả hoặc một nhóm người dùng cụ thể.
* Firebase Storage: Dịch vụ lưu trữ.

**CHƯƠNG 3: PHÂN TÍCH ĐỀ TÀI VÀ ỨNG DỤNG ĐẶT XE**

## **3.1 Đặc tả bài toán**

### *3.1.1 Tổng quan về bài toán đặt xe*

Bài toán đặt xe là một hệ thống ứng dụng di động kết nối giữa người dùng và tài xế, nhằm cung cấp dịch vụ vận chuyển cá nhân thuận tiện và nhanh chóng. Hệ thống này cho phép người dùng tìm kiếm, đặt và thanh toán cho các chuyến đi, trong khi tài xế có thể nhận và quản lý các đơn hàng từ người dùng. Dưới đây là mô tả chi tiết về các phần của hệ thống:

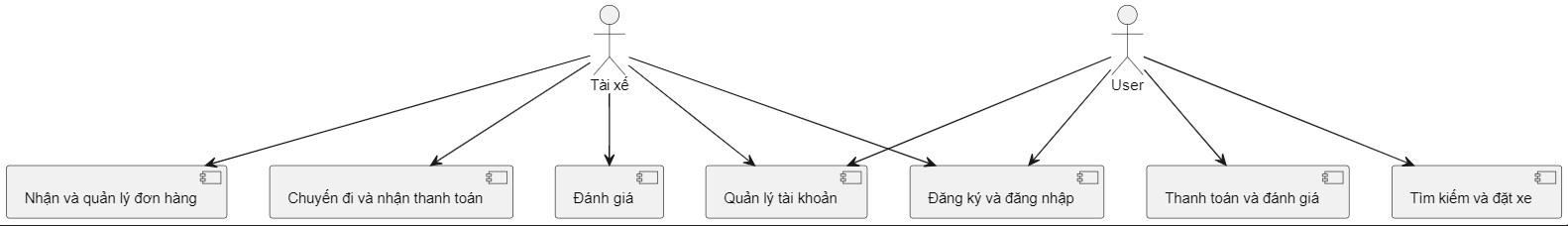
Người dùng:

* Đăng ký và đăng nhập: Người dùng có thể đăng ký tài khoản mới bằng cách cung cấp thông tin cá nhân cần thiết và đăng nhập, đăng xuất vào ra hệ thống để sử dụng các chức năng.
* Quản lý tài khoản: Người dùng có thể xem thông tin cá nhân, bao gồm tên, số điện thoại, địa chỉ và phương thức thanh toán.
* Tìm kiếm và đặt xe: Người dùng có khả năng tìm kiếm điểm đi và điểm đến bằng cách nhập thông tin hoặc chọn trực tiếp trên bản đồ. Hệ thống sẽ tính toán khoảng cách và giá tiền cho từng loại xe (ví dụ: bike, uber-go, uber-x). Sau khi lựa chọn, người dùng có thể gửi yêu cầu chuyến đi (ride request).
* Xem thông tin tài xế: Nếu yêu cầu chuyến đi được chấp nhận, người dùng sẽ được hiển thị thông tin về tài xế, bao gồm tên, hình ảnh và đánh giá của tài xế đó.
* Thanh toán và đánh giá: Sau khi hoàn thành chuyến đi, người dùng có thể thanh toán tiền dịch vụ thông qua phương thức thanh toán trước đó đã lưu trong tài khoản. Họ cũng có thể đánh giá tài xế bằng cách cung cấp số sao và nhận xét về trải nghiệm của mình.

Tài xế:

* Đăng ký và đăng nhập: Tài xế có thể đăng ký tài khoản mới bằng cách cung cấp thông tin cá nhân cần thiết và đăng nhập,đăng xuất vào ra hệ thống.
* Quản lý tài khoản: Tài xế có khả năng xem và cập nhật thông tin cá nhân, bao gồm tên, số điện thoại, địa chỉ và phương thức thanh toán.
* Nhận và quản lý đơn hàng: Tài xế có thể nhận các đơn hàng từ người dùng dựa trên khoảng cách và đánh giá của mình. Họ có thể chấp nhận hoặc từ chối đơn hàng và nếu chấp nhận, họ sẽ được hiển thị thông tin về điểm đến và điểm đón của khách hàng.
* Chuyến đi và thanh toán: Sau khi chấp nhận đơn hàng, tài xế sẽ bắt đầu chuyến đi và đưa khách hàng đến điểm đến. Họ có thể theo dõi tuyến đường được chọn trên ứng dụng. Sau khi hoàn thành chuyến đi, tài xế có thể ghi lại số tiền nhận được và xem lịch sử chuyến đi của mình.
* Đánh giá: Tài xế có thể xem tổng đánh giá sao của mình.

### *3.1.2 Sơ đồ mô tả hệ thống*



Hình 3.1.2 Sơ đồ mô tả hệ thống

*3.1.3 Xác định yêu cầu chức năng và phi chức năng của hệ thống.*

Yêu cầu chức năng:

* Đăng ký và đăng nhập: Người dùng và tài xế có thể đăng ký tài khoản mới và đăng nhập vào hệ thống.
* Quản lý tài khoản: Người dùng và tài xế có thể xem và cập nhật thông tin cá nhân của mình, bao gồm tên, số điện thoại, địa chỉ và phương thức thanh toán.
* Tìm kiếm và đặt xe: Người dùng có khả năng tìm kiếm điểm đi và điểm đến, chọn loại xe và gửi yêu cầu chuyến đi. Tài xế có thể nhận và quản lý các đơn hàng từ người dùng.
* Hiển thị thông tin: Người dùng có thể xem thông tin tài xế như tên, hình ảnh và đánh giá. Tài xế có thể xem thông tin điểm đến và điểm đón của khách hàng.
* Thanh toán: Người dùng có thể thanh toán tiền dịch vụ thông qua phương thức thanh toán trước đó đã lưu trong tài khoản.
* Đánh giá: Người dùng có thể đánh giá tài xế sau mỗi chuyến đi bằng cách cung cấp số sao và nhận xét. Tài xế cũng có thể xem thông tin đánh giá từ người dùng.

Yêu cầu phi chức năng:

* Giao diện người dùng thân thiện: Hệ thống cần cung cấp giao diện người dùng dễ sử dụng và thân thiện để người dùng và tài xế có thể dễ dàng tương tác với các chức năng.
* Bảo mật thông tin: Hệ thống cần đảm bảo an toàn và bảo mật thông tin cá nhân của người dùng và tài xế.
* Hiệu suất: Hệ thống cần đảm bảo hoạt động ổn định và đáp ứng nhanh chóng các yêu cầu của người dùng và tài xế.
* Độ tin cậy: Hệ thống cần đảm bảo tính tin cậy trong việc đưa ra thông tin về tài xế, đánh giá và lịch sử chuyến đi.
* Tương thích đa nền tảng: Hệ thống cần tương thích với các nền tảng di động phổ biến như Android và iOS để người dùng và tài xế có thể truy cập từ các thiết bị khác nhau.

*3.1.4 Định nghĩa, khái niệm và thuật ngữ liên quan trong lĩnh vực đặt xe.*

* Người dùng: Người dùng là người sử dụng ứng dụng để tìm kiếm, đặt và thanh toán cho dịch vụ vận chuyển cá nhân.
* Tài xế: Tài xế là người cung cấp dịch vụ vận chuyển cho người dùng và nhận các đơn hàng từ hệ thống.
* Điểm đi: Điểm đi là vị trí mà người dùng muốn được đón để bắt đầu chuyến đi.
* Điểm đến: Điểm đến là vị trí mà người dùng muốn đến sau khi hoàn thành chuyến đi.
* Loại xe: Loại xe đại diện cho các phương tiện vận chuyển có sẵn để người dùng lựa chọn, ví dụ: bike, uber-go, uber-x.
* Yêu cầu chuyến đi (ride-request): Yêu cầu chuyến đi là thông tin được người dùng gửi đến hệ thống để yêu cầu vận chuyển từ điểm đi đến điểm đến. Nó bao gồm thông tin về điểm đi, điểm đến, loại xe và các yêu cầu khác (nếu có).
* Thanh toán: Thanh toán là quá trình người dùng thanh toán tiền dịch vụ sau khi hoàn thành chuyến đi. Phương thức thanh toán có thể là ví điện tử, thẻ tín dụng hoặc các phương thức khác mà người dùng đã đăng ký trước đó.
* Đánh giá: Đánh giá là quá trình người dùng hoặc tài xế đánh giá trải nghiệm của họ sau mỗi chuyến đi. Nó thường được thể hiện dưới dạng số sao và nhận xét về chất lượng dịch vụ, tài xế và các khía cạnh khác.

## **3.2 Phát triển ứng dụng**

### *3.2.1 Quy trình phát triển, thiết kế giao diện ứng dụng.*

*3.2.2 Triển khai và kiểm thử.*

# **CHƯƠNG 4: KẾT LUẬN VÀ KIẾN NGHỊ**

**4.1 Kết quả đạt được:**

Trong đề tài này, tôi đã xây dựng thành công một ứng dụng đặt xe thông minh sử dụng công nghệ Flutter, Flutter Map và Firebase. Các kết quả đạt được bao gồm:

* Thiết kế và phát triển giao diện người dùng thân thiện, cho phép người dùng dễ dàng tìm kiếm và đặt xe thông qua ứng dụng.
* Tích hợp Flutter Map để hiển thị bản đồ và vị trí xe đặt sẵn, giúp người dùng dễ dàng xác định vị trí và chọn xe gần nhất.
* Sử dụng Firebase để lưu trữ và quản lý dữ liệu người dùng, xe và các giao dịch đặt xe.
* Cung cấp chức năng đặt và hủy xe thông qua giao diện người dùng đơn giản và trực quan.
* Xử lý thanh toán và gửi thông báo xác nhận đến người dùng sau khi đặt xe thành công.

**4.2 Hạn chế tồn tại:**

Trong quá trình phát triển, tôi đã gặp một số hạn chế, bao gồm:

* Hiệu năng: Đôi khi ứng dụng có thể gặp vấn đề về hiệu năng khi tải dữ liệu từ Firebase hoặc hiển thị các đối tượng trên bản đồ.
* Bảo mật: Mặc dù tôi đã sử dụng Firebase để xác thực và quản lý người dùng, nhưng việc bảo mật dữ liệu và giao tiếp vẫn cần được cải thiện để đảm bảo an toàn cho người dùng và dữ liệu cá nhân.

**4.3 Hướng phát triển:**

Dựa trên kết quả đạt được và hạn chế tồn tại, tôi đề xuất các hướng phát triển tiềm năng cho ứng dụng đặt xe thông minh:

* Tối ưu hiệu năng: Nâng cao hiệu năng của ứng dụng bằng cách tối ưu hóa việc tải dữ liệu, caching và áp dụng các biện pháp tăng tốc khác.
* Mở rộng phạm vi: Mở rộng ứng dụng để hỗ trợ nhiều thành phố và khu vực hơn, cung cấp nhiều tùy chọn xe và địa điểm đặt xe cho người dùng.
* Tích hợp tính năng thời gian thực: Thêm tính năng theo dõi vị trí thời gian thực của xe và cập nhật trạng thái đặt xe để người dùng có thể xem thông tin và thay đổi lựa chọn dựa trên tình trạng hiện tại.
* Nâng cao bảo mật: Tăng cường bảo mật ứng dụng bằng cách sử dụng giao thức HTTPS, mã hóa dữ liệu và xác thực hai yếu tố để đảm bảo rằng thông tin cá nhân của người dùng được bảo vệ một cách an toàn.
* Tích hợp hệ thống thanh toán: Nâng cấp hệ thống thanh toán để hỗ trợ nhiều các phương thức thanh toán phổ biến như ví điện tử, thẻ tín dụng và ví điện tử.

## **4.4 Kết luận:**

Trong tổng quan, việc xây dựng ứng dụng đặt xe thông minh sử dụng Flutter, Flutter Map và Firebase đã đạt được những kết quả tích cực. Ứng dụng cung cấp cho người dùng trải nghiệm đơn giản và thuận tiện để đặt xe thông qua giao diện người dùng thân thiện. Tuy nhiên, còn nhiều khả năng để phát triển và nâng cao ứng dụng trong tương lai.

Để đạt được sự thành công và sự phát triển bền vững cho ứng dụng đặt xe thông minh, tôi đề xuất tiếp tục nghiên cứu và phát triển các tính năng mới, cải thiện hiệu năng và bảo mật, mở rộng phạm vi hoạt động và tích hợp các phương thức thanh toán. Đồng thời, liên tục lắng nghe ý kiến và phản hồi từ người dùng để cải thiện trải nghiệm sử dụng và đáp ứng nhu cầu của họ.

Với sự phát triển và cải tiến liên tục, ứng dụng đặt xe thông minh có tiềm năng trở thành một giải pháp hữu ích và tiện lợi cho việc di chuyển trong thành phố và góp phần vào sự phát triển của ngành công nghiệp giao thông.

Tuy có rất nhiều ưu điểm nhưng cũng không tránh khỏi những sai sót. Vì vậy cần thêm nhiều thời gian nghiên cứu và thực hành để cải tiến phần mềm tốt hơn.

Mặc dù đã rất cố gắng trong việc nghiên cứu và thực hiện đồ án, nhưng do thời gian và hiểu biết của tôi vẫn còn hạn chế nên đồ án chỉ dừng lại ở mức ứng dụng đơn giản mà chưa kịp thực hiện và triển khai ứng dụng vào thực tế bởi hiện nay sẽ cần thêm rất nhiều kiến thức cũng như hiểu biết ở các lĩnh vực công nghệ. Đồng thời, đồ án chắc chắn cũng không tránh khỏi những thiếu sót, nên tôi rất mong nhận được ý kiến đóng góp từ các thầy cô .

**TÀI LIỆU THAM KHẢO**

[1] Flutter documentation. Truy cập: <https://flutter.dev/docs>

[2] Firebase documentation. Truy cập: <https://firebase.google.com/docs>

[3] Geocoding API documentation. Truy cập: <https://developers.google.com/maps/documentation/geocoding/overview>

[4] Firebase Cloud Messaging (FCM) documentation. Truy cập: <https://firebase.google.com/docs/cloud-messaging>

[5] Chen, Y., Wang, X., and Wang, H. "Design and Implementation of a Taxi Hailing System Based on Mobile Internet." In 2015 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), pages 191-194. IEEE, 2015.

[6] Li, K., Zhang, L., and Zhang, Z. "A Real-Time Taxi Hailing System Based on Mobile Internet." In 2014 11th International Conference on Service Systems and Service Management, pages 40-45. IEEE, 2014.

[7] Wang, W., Li, X., and He, Y. "Design and Implementation of Taxi Hailing System Based on Android." In 2017 2nd International Conference on Automation, Control and Robotics Engineering (CACRE), pages 97-101. IEEE, 2017.

[8] Liu, X., Yang, Y., and Liu, Y. "Design and Implementation of a Mobile Taxi Hailing System Based on WeChat Public Platform." In 2018 13th International Conference on Computer Science & Education (ICCSE), pages 14-18. IEEE, 2018.

[9] Wang, Y., Xu, Y., and Li, H. "Design and Implementation of a Taxi Hailing System Based on GPS and GIS." In 2016 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), pages 227-231. IEEE, 2016.