**ĐẠI HỌC QUỐC GIA HÀ NỘI**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

**TRẦN VĂN THẮNG**

**XÂY DỰNG ỨNG DỤNG ĐI CHUNG XE**

**(PHÍA CHỦ XE)**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành: Công nghệ thông tin**

**Cán bộ hướng dẫn: *TS. Lê Hồng Hải***

## HÀ NỘI - 2019

## HÀ NỘI - 2019

## HÀ NỘI - 2019

## HÀ NỘI - 2019

## HÀ NỘI - 2019

## HÀ NỘI - 2019

## HÀ NỘI - 2019

## HÀ NỘI - 2019

## HÀ NỘI - 2019

## HÀ NỘI - 2019

## HÀ NỘI - 2019

## HÀ NỘI - 2019

## HÀ NỘI - 2019

## HÀ NỘI - 2019

## HÀ NỘI - 2019

## HÀ NỘI - 2019

**TÓM TẮT NỘI DUNG**

**Tóm tắt**: Những năm gần đây tình trạng tắc đường tại hầu hết các đô thị lớn ở trên thế giới và trong đó có Việt Nam đặc biệt vào các ngày chuẩn bị nghỉ lễ khi mà mọi người lên kế hoạch về với gia đình đã trở thành một vấn đề mà tất cả mọi người đều quan tâm. Tình trạng này ít nhiều đã làm ảnh hưởng tiêu cực đến nhiều mặt của đời sống xã hội như: ô nhiễm môi trường do lãng phí nhiên liệu, gây lãng phí thời gian cho mọi người, gây hại sức khỏe cho cộng đồng, gây thiệt hai cho nền kinh tế của đất nước,… Mặc dù nhà nước đã đưa ra rất nhiều biện pháp nhằm giảm ùn tắc giao thông nhưng những biện pháp đó giường như chưa có hiệu quả. Một trong những nguyên nhân dẫn đến tình trạng này là do sự gia tăng chóng mặt của các phương tiện cá nhân như ô tô, xe máy. Khi quan sát tình trạng này ta có thể thấy số lượng người lái xe một mình trong khi còn rất nhiều chỗ trống là rất nhiều, điều này là vô cùng lãng phí.

Kèm theo đó có thể thấy điện thoại thông minh hay smartphone trong những năm gần đây đang dần trở nên phổ biến tới cộng đồng xã hội. Từ tuổi teen, thanh niên, trung niên, người già hầu như ai cũng sở hữu trên tay một chiếc điện thoại thông minh để sử dụng hàng ngày. Cùng với sự phát triển của mạng Internet đã giúp mọi người trong cộng đồng có thể kết nối với nhau một cách tiện lợi, nhanh chóng hơn.

Với mục đích giải quyết một phần vế đề tắc đường, tôi muốn xây dựng một ứng dụng đi chung xe trên nền tảng di động giúp kết nối các chủ phương tiện có ghế trống với hàng khách muốn tìm kiếm một chuyến đi nào đó. Ứng dụng với mục tiêu giúp những người có phương tiện cá nhân có thể chia sẻ chuyến đi của mình và những người khác có thể vào và tìm chuyến đi cùng đường với chủ phương tiện một cách thuận tiện, tiết kiệm và chia sẻ một phần chi phí với chủ phương tiện.

***Từ khóa:*** Tắc đường, Đi chung xe.

**TÀI LIỆU THAM KHẢO**

**LỜI CAM ĐOAN**

Tôi xin cam đoan khóa luận tốt nghiệp do tôi tự thực hiện dưới sự hướng dẫn của Thầy Lê Hồng Hải, mọi thông tin tham khảo sử dụng trong khóa luận đều đã được trích dẫn đầy đủ và hợp pháp.

Tôi xin hoàn toàn chịu trách nhiệm và chịu mọi hình thức kỷ luật theo quy định của nhà trường cho lời cam đoan của mình.

Hà Nội, ngày tháng 5 năm 2019

Người cam đoan

**LỜI CẢM ƠN**

**MỤC LỤC**

[**CHƯƠNG 1. GIỚI THIỆU VỀ BÀI TOÁN** 9](#_Toc5022171)

[**1.1.** **Bối cảnh và nhu cầu thực tế** 9](#_Toc5022172)

[**1.2.** **Giải quyết vấn đề** 12](#_Toc5022173)

[**CHƯƠNG 2. KIẾN TRÚC HỆ THỐNG VÀ PHÁT TRIỂN ỨNG DỤNG** 13](#_Toc5022174)

[2.1. Kiến trúc hệ thống 13](#_Toc5022175)

[**2.1.1. Client** 13](#_Toc5022176)

[**2.1.2. Firebase** 13](#_Toc5022177)

[**2.1.3. Server** 14](#_Toc5022178)

[2.2. Tổng quan về công nghệ 14](#_Toc5022179)

[**2.2.1. NativeScript** 14](#_Toc5022180)

[**2.2.1.1. NativeScript là gì?** 14](#_Toc5022181)

[**2.2.1.2. Nền tảng kiến trúc của ứng dụng NativeScript** 14](#_Toc5022182)

[**2.2.1.3. Sơ đồ kiến trúc của NativeScript** 15](#_Toc5022183)

[**2.2.2. Angular** 16](#_Toc5022184)

[**2.2.2.1. TypeScript** 16](#_Toc5022185)

[**2.2.2.2. Khái niệm về Angular** 17](#_Toc5022186)

[**2.2.2.3. Kiến trúc của Angular** 17](#_Toc5022187)

[**2.2.2.3.1. Modules** 18](#_Toc5022188)

[**2.2.2.3.2. Components** 19](#_Toc5022189)

[**2.2.2.3.3. Templates, directives, and data binding** 19](#_Toc5022190)

[**2.2.2.3.4. Services and dependency injection** 19](#_Toc5022191)

[**2.2.2.3.5 Routing** 20](#_Toc5022192)

[**2.2.3. Web Services** 20](#_Toc5022193)

[**2.2.3.1. Giới thiệu về Web Services** 20](#_Toc5022194)

[**2.2.3.2. RESTful Web Services** 20](#_Toc5022195)

[**2.2.3.2.1. Kiến trúc REST** 20](#_Toc5022196)

[**2.2.4. Laravel** 21](#_Toc5022197)

[**2.2.4.1. Giới thiệu** 21](#_Toc5022198)

[**2.2.4.2. Các tính năng của Laravel** 22](#_Toc5022199)

[**2.2.4.2.1. Truy cập cơ sở dữ liệu** 22](#_Toc5022200)

[**2.2.4.2.2. Công cụ dòng lệnh** 22](#_Toc5022201)

[**2.2.4.2.3. Định tuyến** 23](#_Toc5022202)

[**2.2.4.2.4. View Template engine** 23](#_Toc5022203)

[**2.2.4.2.5. Lên kế hoạch cho công việc** 23](#_Toc5022204)

[**2.2.4.2.6. Hệ thống tệp tin (File System)** 23](#_Toc5022205)

[**2.2.4.2.7. Xác thực** 23](#_Toc5022206)

[**2.2.5. Firebase** 23](#_Toc5022207)

[**2.2.5.1. Giới thiệu về Firebase** 23](#_Toc5022208)

[**2.2.5.2. Một số dịch vụ của Firebase** 24](#_Toc5022209)

[**2.2.5.2.1. Cloud Firestore** 24](#_Toc5022210)

[**2.2.5.2.2. ML Kit** 24](#_Toc5022211)

[**2.2.5.2.3. Cloud Funtions** 25](#_Toc5022212)

[**2.2.5.2.4. Authentication** 25](#_Toc5022213)

[**2.2.5.2.5. Hosting** 25](#_Toc5022214)

[**2.2.5.2.6. Cloud Storage** 25](#_Toc5022215)

[**2.2.5.2.7. Realtime Database** 25](#_Toc5022216)

[**2.2.5.2.8. Crashlytics** 26](#_Toc5022217)

[**2.2.5.2.9. Performance Monitoring** 26](#_Toc5022218)

[**2.2.5.2.10. Test Lab** 26](#_Toc5022219)

[**2.2.5.2.11. In-App Messaging** 26](#_Toc5022220)

[**2.2.5.2.12. Google Analytics** 26](#_Toc5022221)

[**2.2.5.2.13. Predictions** 26](#_Toc5022222)

[**2.2.5.2.14. A/B Testing** 27](#_Toc5022223)

[**2.2.5.2.15. Cloud Messaging** 27](#_Toc5022224)

[**2.2.5.2.16. Remote Config** 27](#_Toc5022225)

[**2.2.5.2.17. Dynamic Links** 27](#_Toc5022226)

[**2.2.5.2.18. App Indexing** 27](#_Toc5022227)

[**2.2.6. Google Maps Api** 28](#_Toc5022228)

[**2.2.6.1. Giới thiệu về Google Maps** 28](#_Toc5022229)

[**2.2.6.2. Google Maps Api** 28](#_Toc5022230)

[2.3. Xác định yêu cầu 29](#_Toc5022231)

[**2.3.1. Yêu cầu của ứng dụng** 29](#_Toc5022232)

[**2.3.2. Đặc tả yêu cầu** 29](#_Toc5022233)

[**2.3.2. Yêu cầu cụ thể** 30](#_Toc5022234)

[**2.3.2.1. Yêu cầu chức năng** 30](#_Toc5022235)

[**2.3.2.2. Yêu cầu phi chức năng** 31](#_Toc5022236)

[2.3. Mô hình ca sử dụng 32](#_Toc5022237)

# **CHƯƠNG 1. GIỚI THIỆU VỀ BÀI TOÁN**

* 1. **Bối cảnh và nhu cầu thực tế**

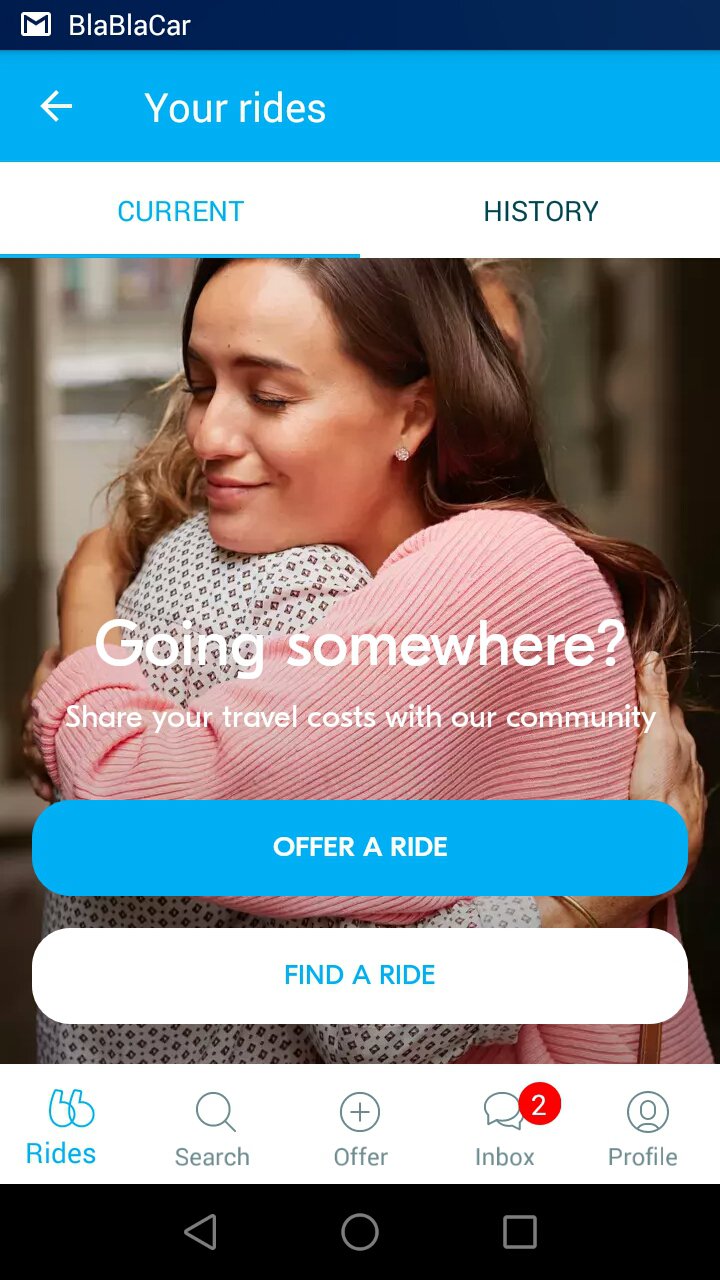
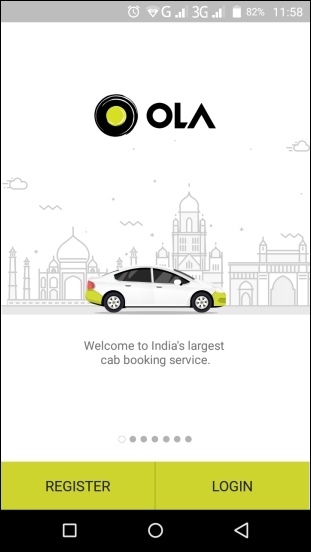
Giao thông là một lĩnh vực quan trọng đối với bất kỳ nước nào trên thế giới. Nhờ có hệ thống giao thông phát triển mà mọi người có thể di chuyển một cách dễ dàng, thuận lợi, hàng hóa được vận chuyển đến mọi nơi. Hàng năm nhà nước Việt Nam đã đầu tư rất nhiều để đưa ra những phương án để giải quyết những vấn đề nhức nhối của giao thông hiện tại như tắc đường, an toàn giao thông,…



**Hình 1.1. Tình trạng tắc đường ở Việt Nam**

Trong đó tình trạng tắc đường tại các thành phố lớn từ nhiều năm nay luôn là một vấn đề nhức nhối mà cả xã hội luôn quan tâm, không chỉ ở Việt Nam mà còn trên toàn thế giới, đặc biệt là vào các ngày lễ, tết và các giờ cao điểm. Vào những giờ cao điểm hay trước các dịp nghỉ lễ, tết trong khi các phương tiện như xe bus, tàu luôn trong tình trạng chật cứng, hết vé thì tại các ngã ba, ngã tư, các điểm, con đường mấu chốt hầu như lúc nào cũng xảy ra tình trạng ùn tắc kéo dài hàng giờ đồng hồ liền. Tình trạng này gây ảnh hưởng tiêu cực đến mọi mặt của đời sống xã hội như: ô nhiễm môi trường, lãng phí thời gian, nhiên liệu, hàng hóa lưu thông chậm ảnh hưởng tiêu cực đến nền kinh tế,... Ở Việt Nam một trong nhữn nguyên nhân chính dẫn đến tình trạng này là do sự gia tăng của các phương tiện cá nhân ngày càng cao, có thể thấy một cá nhân có thể sở hữu nhiều phương tiện xe máy, ô tô, hơn thế nữa khi quan sát những phương tiện cá nhân này khi tham gia giao thông có thể thấy đa số các phương tiện còn nhiều ghế trống trong khi họ phải len lách qua những chỗ bị ùn tắc một cách khó khăn.

Trên thế giới, kéo theo sự phát triển của Internet và xu hướng sử dụng các thiết bị di động thông minh trên nhiều nền tảng như IOS, Android đã có nhiều ứng dụng được xây dựng cũng nhằm mục địch giải quyết vấn đề này như BlaBlaCar của công ty Comuto, Blacklane của Blacklane GmbH, Ola Cabs,… Điển hình là hệ thống BlaBlaCar bao của công ty Comuto đã được ra mặt ở hầu hết các quốc gia Châu Âu với khoảng 70 triệu người dùng vào năm 2019. Các ứng dụng này đã góp phần không nhỏ vào việc giải quyết vấn đề tắc đường ở hầu hết các quốc gia mà nó có mặt.



**Hình 1.2. Một số ứng dụng đi chung xe trên thế giới**

Hiện nay ở Việt Nam Internet cũng vô cùng phát triển và được phổ biến rộng rãi tới mọi người, kéo theo đó là sự phổ biến rộng rãi các thiết bị di động thông mình trong cộng đồng người Việt Nam. Thông qua môi trường Internet, mọi người có thể kết nối với nhau một cách thuận tiện, dễ dàng hơn. Nhận thấy ta có thể xây dựng một ứng dụng đi chung xe trên thiết bị di động mà hầu như ai cũng sở hữu cho người Việt để giải quyết một phần nào đó vấn đề tắc đường đang nhức nhối này ở Việt Nam.

* 1. **Giải quyết vấn đề**

Thiết kế và xây dựng một ứng dụng trên nền tảng di động theo mô hình máy khách-máy chủ có thể giải quyết một phần nào đó tình trạng tắc đường hiện nay. Ứng dụng đi chung xe này tạo nên một cộng đồng giúp kết nối mọi người với nhau cụ thể là kết nối giữa những người lái xe với phương tiện cá nhân của mình với ghế trống với những người khác có cùng mong muốn đi tới một địa điểm nào đó, thông qua đó 2 người có thể trao đổi với nhau để người lái xe có thể chia sẻ ghế trống của mình cho khách hàng và khách hàng sau khi đi có thể chia sẻ một phần chi phí với chủ phương tiện.

Người dùng cuối của hệ thống gồm:

Chủ phương tiện: Những người có phương tiện muốn chia sẻ chuyến đi của mình với mọi người.

Hàng khách: Những người muốn tìm một chuyến đi nào đó cho mình.

Trong hệ thống này, một người dùng vừa có thể là hàng khách vừa có thể là chủ xe khi đăng ký thông tin của mình.

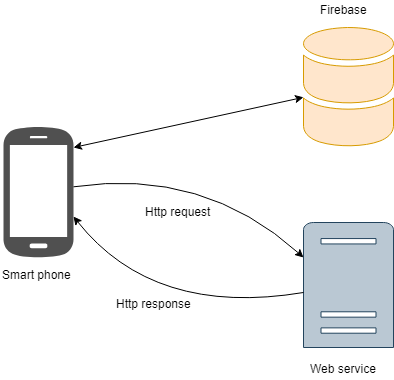
Với người là chủ phương tiện có thể cung cấp thông tin một cách công khai nên hệ thống, khi có nhu cầu đi đâu đó mà muốn chia sẻ chuyến đi với ghế trống với mọi người thì có thể truy cập để tạo cho mình lịch trình một chuyến đi với thời gian, địa điểm cụ thể, sau đó có thể đăng thông tin lên hệ thống một cách công khai với mọi người trong cộng đồng của ứng dụng.

Còn với hàng khác - người muốn tìm một chuyến đi cho mình thì có thể truy cập ứng dụng rồi nhập thông tin về lịch trình mà mình muốn tìm kiếm, khi đó hệ thống sẽ tự động gợi ý ra những chuyến đi phù hợp với thông tin mà hàng khách tìm kiếm. Sau đó người dùng có thể lựa chọn chuyến đi thích hợp cho mình và trao đổi trực tiếp với chủ xe để đưa ra hỏa thuận với chủ xe về chuyến đi. Sau khi chuyến đi hoàn thành thì khách hàng có thể chia sẻ một phần chi phí đã được thương lượng từ trước với phía chủ phương tiện và đánh giá độ tin cậy của chủ phương tiện đó.

# **CHƯƠNG 2. KIẾN TRÚC HỆ THỐNG VÀ PHÁT TRIỂN ỨNG DỤNG**

## 2.1. Kiến trúc hệ thống

Ứng dụng được xây dựng theo mô hình client-server



**Hình 2.1. Mô hình kiến trúc hệ thống**

### **2.1.1. Client**

Ứng dụng được xây dựng dành cho các thiết bị di động hệ điều hành Android trên nền tảng NativeScript chịu trách nhiệm giao tiếp với người dùng là các tài xế và khách hàng và gửi những yêu cầu tới server và nhận kết quả trả về từ server rồi xử lý chúng.

### **2.1.2. Firebase**

Hệ thống sử dụng nền tảng phát triển di động của Goolge là Firebase với mục đích:

* Quản lý những người dùng của hệ thống dành cho việc xác thực, đăng ký, đăng nhập.
* Sử dụng dịch vụ cơ sở dữ liệu thời gian thực (Realtime Database) để lưu trữ những nội dung tin nhắn của người dùng hệ thống.
* Sử dụng dịch vụ đẩy thông báo của firebase để đẩy mọi thông báo từ hệ thống đến từng người dùng cụ thể.

### **2.1.3. Server**

Phần server sử dụng công nghệ dịch vụ web (Webservices) với kiến trúc REST được xây dựng trên framework laravel. Server sử dụng RESTful API để giao tiếp với phía client, thông qua các API này mà server nhận yêu cầu được gửi từ phía client rồi tiến hành xử lý và trả về các kết quả cho client.

## 2.2. Tổng quan về công nghệ

### **2.2.1. NativeScript**

#### **2.2.1.1. NativeScript là gì?**

Hiện nay trên thế giới số lượng các thiết bị di động với các hệ điều hành phổ biến như IOS, Android,… mà con người sử dụng đang là rất lớn. Điều này đã tạo nên một hệ sinh thái di động vô cùng tiềm năng mà đa số doanh nghiệp hay các nhà phát triển đều mong muốn chinh phục nó. Với các nhà phát triển khi xây dựng một ứng dụng nào đó luôn muốn ứng dụng đó tiếp tới nhiều người dùng với nhiều nền tảng hỗ trợ để có thể thu hút nhiều người dùng càng tốt và với chất lượng tốt nhất. Đối với nhiều ứng dụng trước đây khi ta muốn xây dựng nó trên nhiều nền tảng khác nhau thì đòi hỏi lập trình viên ứng dụng di động phải biết nhiều ngôn ngữ tương ứng để xây dựng ứng dụng cho từng nền tảng hay cần phải có sự phối hợp của một đội ngũ để xây dựng trên các nền tảng riêng với chuyên môn của họ, điều này sẽ dẫn đến cần nhiều thời gian và chi phí để có thể xây dựng ứng dụng. Hơn thế nữa khi mà các lập trình viên website cũng muốn phát triển một ứng dụng di động thì hướng tiếp cận với họ ban đầu là tương đối khó. Để giải quyết điều đó hiện nay đã ra đời nhiều framework cho phép các lập trình viên có thể xây dựng một ứng dụng di động đa nền tảng trên cả IOS và Android. Mộ trong những framework đó là Nativescript được xây dựng và phát triển bới Telerik.

NativeScript là một framework ra đời dùng để xây dựng các dụng đa nền tảng cho cả IOS và Android, các lập trình viên chỉ cần viết một lần là có thể chạy trên cả 2 nền tảng này. NativeScript sử dụng các công nghệ web như Angular, Vue.js, Typescript, Flexbox và CSS để có một trải nghiệm giao diện người dùng thực sự khi sử dụng lại các kỹ năng và mã nguồn từ các dự án web của bạn. Nhận 100% quyền truy cập vào các API gốc thông qua JavaScript và sử dụng lại các gói từ npm, CocoaPods and Gradle. Đây là một mã nguồn mở và miễn phí.

#### **2.2.1.2. Nền tảng kiến trúc của ứng dụng NativeScript**

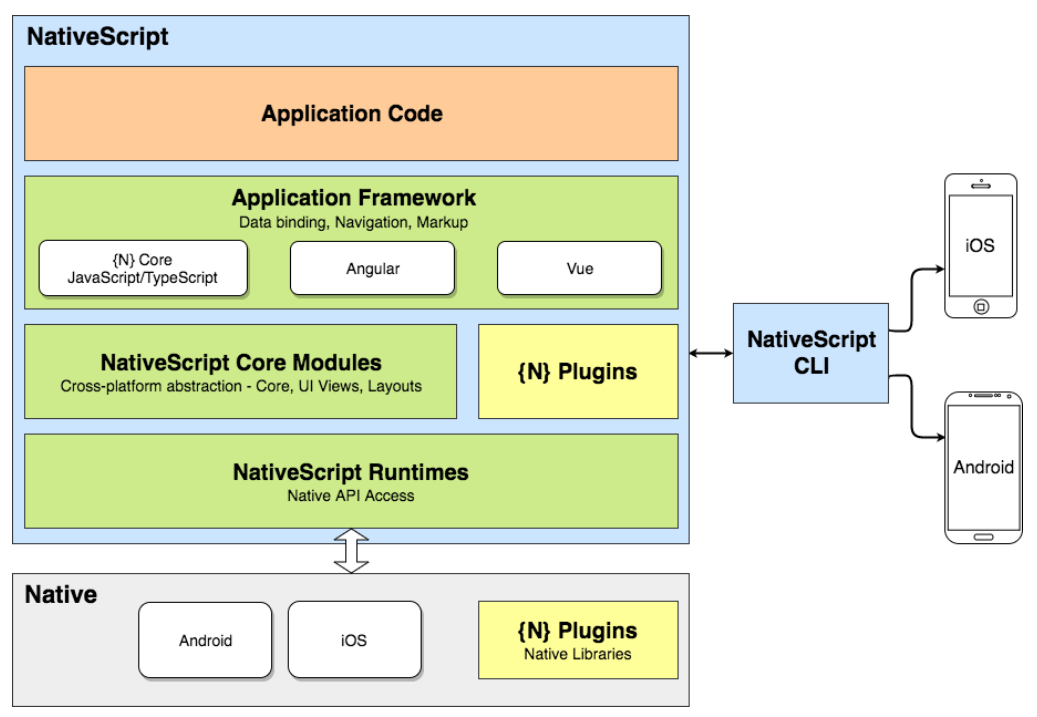
Dù NativeScript được xây dựng dựa trên các công nghệ web như Angular, Vue.js, Typescript, Flexbox và CSS nhưng bạn có thể xây dựng được các ứng dụng đa nền tảng cho phép tạo các ứng dụng nguyên gốc Native Application. Do đó các ứng dụng bằng NativeScript sẽ mang lại hiệu năng hơn hẳn các ứng dụng Hybrid Application.

Các ứng dụng Hybrid Application là các ứng dụng đa nền tảng được lưu bên trong một ứng dụng NativeScript Application có sử dụng Webview được xây dựng bằng các công nghệ web như HTML, CSS, Javascript. Chính vì vậy mà các ứng dụng này có thể truy cập các chức năng của phần cứng thông qua một tập các giao diện ứng dụng API javascript được xây dựng bằng native code.

NativeScript Application là những ứng dụng được các nhà phát triển viết riêng cho một nền tảng nhất định nào đó như IOS, Android,… bằng các ngôn ngữ tương ứng với từng nền tảng. Chính vì vậy các ứng dụng NativeScript Application sẽ mang lại một hiệu năng hơn hẳn các ứng dụng Hybrid Application trong hầu hết hoàn cảnh. Hơn thế các ứng dụng này có khả năng khai thác triệt để các sức mạnh phần cứng như camera, gps, cảm biến,… và các quyền local như đọc ghi các tập tin, truy cập tin nhắn, danh bạ,…

NativeScript có thể cung cấp trải nghiệm phát triển tốt nhất trên cả 2 nền tảng. NativeScript sử dụng máy ảo JavaScript để thông dịch và thực thi mã nguồn JavaScript để dịch thành các lời gọi trực tiếp tới các giao diện API gốc của hệ thống. Nhờ đó mà các modules JavaScipt đa nền tảng của NativeScript cung cấp cho các nhà phát triển nhiều tiện lợi khi viết ứng dụng di động qua khả năng truy cập các API, SDK và các framework khi cần.

#### **2.2.1.3. Sơ đồ kiến trúc của NativeScript**



**Hình 2.2. Sơ đồ kiến trúc của NativeScript**

Công nghệ NativeScript được xây dựng dựa trên một số thành phần chính:

* Runtimes: Cho phép bạn gọi API trong khung Android và IOS bằng mã JavaScript bằng cách sử dụng máy ảo JavaScript - V8 của Google cho Android và triển khai JavaScriptKore của WebKit được phân phối với iOS 7.0+
* Core Modules: cung cấp các trừu tượng cần thiết để truy cập các nền tảng gốc bên dưới.Ví dụ như modules Gestures định nghĩa một API JS phổ biến để dịch mã TypeScript/JavaScript của ứng dụng thành các lệnh gọi API Gestures gốc (nhờ vào Runtimes). Bên cạnh đó các Core Modules dựa trên XML cơ bản để xác định các UI, liên kết dữ liệu và điều hướng
* NativeScript CLI: Giao diện dòng lệnh cho phép tạo, xây dựng và chạy các ứng dụng bằng NativeScript CLI
* NativeScript Plugins: Xây dựng các khối đóng gói một số chức năng và giúp các nhà phát triển xây dựng ứng dụng nhanh hơn. Hầu hết là do cộng đồng xây dựng, được viết bằn TypeScript/JavaScript. Một số có thể bao gồm các thư viện riêng, được gọi từ mã TypeScript/JavaScript nhờ Runtimes

### **2.2.2. Angular**

#### **2.2.2.1. TypeScript**

TypeScript là một dự án mã nguồn mở được phát triển bởi Microsoft, được coi là một phiên bản nâng cao của JavaScript nó được bổ sung những tùy chọn kiểu tĩnh và lớp trên cơ sở lập trình hướng đối tượng mà ở JavaScript không có. TypeScript bao hàm luôn cả ES6(ECMAScript 6 2015) – phiên bản mới nhất của JavaScript, do đó có thể sử dụng tất cả các tính năng của ES6 như namespace, class, module.

TypeScript là ngôn ngữ dành cho JavaScript ở quy mô ứng dụng, nó hỗ trợ các công cụ cho các ứng dụng JavaScript quy mô lớn cho bất kỳ trình duyệt, mọi máy chủ hay bất cứ hệ điều hành nào. Khi phát triển ứng dụng thì TypeScript sẽ được biên dịch thành JavaScript dựa trên mọi tiêu chuẩn để có thể đọc được.

Ưu điểm của TypeScript:

* Khắc phục những nhược điểm hiện có ở JavaScript như cho phép xây dựng mô hình mã nguồn dựa trên mô hình hướng đối tượng, sử dụng các kiểu dữ liệu chặt chẽ cho các thành phần khai báo, có thể được sử dụng để phát triển ứng dụng cho backend (NodeJs) hay frontend với các ứng dụng chạy trên trình duyệt, mobile hay desktop.
* TypeScript được triển khai trên các dự án với quy mô lớn một cách thuận tiện, dễ dàng.
* Là một mã nguồn mở hoàn toàn miễn phí và có một cộng đồng hỗ trợ ngày càng lớn
* TypeScript luôn hỗ trợ các tính năng của phiên bản JavaScript mới nhất và còn hơn thế nữa. TypeScipt khi biên dịch sẽ tạo ra các đoạn mã JavaScript tương ứng nên có thể chạy trên nhiều công cụ có hỗ trợ biên dịch JavaScript.

Hiên nay có rất nhiều framework đã lựa chọn TypeScript để sử dụng như Angular, Ionic,… nhờ những ưu điểm vượt trội của nó.

#### **2.2.2.2. Khái niệm về Angular**

Angular là một framework JavaScript do google phát triển, nó giúp cho các nhà phát triển xây dựng các ứng dụng hiện đại trên nền web, desktop hay các Native Application cho các thiết bị di động.

Angular là phiên bản được phát triển từ AngularJS, nó kế thừa được nhiều ưu điểm từ AngularJS và đã có những cải tiến nhiều làm cho cho nhanh hơn, có khả năng mở rộng hơn và hiện đại hơn

Hơn thế nữa Angular ban đầu đã được thiết kế cho các ứng dụng di động. Angular đã hỗ trợ tạo nên những ứng dụng di động bằng cách tích hợp với NativeScript, một thư viện mã nguồn mỡ giúp kết nối JavaScript và di động.

#### **2.2.2.3. Kiến trúc của Angular**

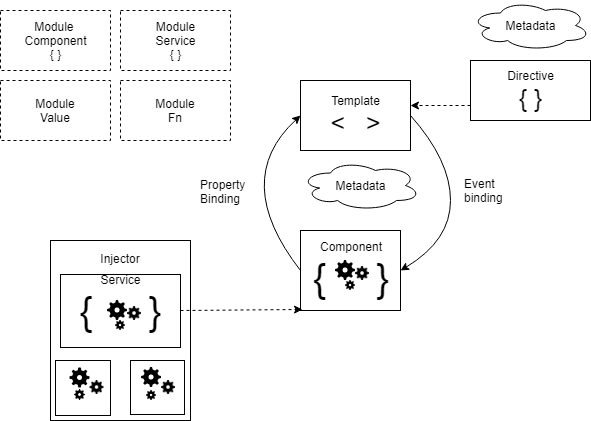
Các khối xây dựng cơ bản của một ứng dụng Angular là NgModules – cung cấp bối cảnh cho các component. NgModules thu thập mã nguồn liên quan vào các bộ chức năng; một ứng dụng Angular được xác đinh bới một bộ NgModules. Một ứng dụng Angular luôn có ít nhất một mô-đun gốc:

* Các component xác định các view – là tập hợp các thành phần màn hình mà Angular có thể chọn trong số đó và sửa đổi theo logic và dữ liệu chương trình của bạn.
* Các component sử dụng các service, cung cấp chức năng cụ thể không liên quan trực tiếp đến các view. Các service có thể được đưa vào các thành phần dưới dạng phụ thuộc, làm cho mã của bạn được mô-đun hóa, có thể được sử dụng lại và hiệu quả.

Cả component và service chỉ đơn giản là các lớp với các trình đánh dấu loại của chúng và cung cấp siêu dữ liệu cho Angular biết cách sử dụng chúng.

* Siêu dữ liệu cho một lớp thành phần liên kết nó với một mẫu xác mà xác định một view. Một mẫu kết hợp HTML với các Angular directive và đánh dấu ràng buộc cho phép Angular chỉnh sửa HTML trước khi hiển thị nó để hiển thị.
* Siêu dữ liệu cho một lớp dịch vụ cung cấp thông tin mà Angular cần để cung cấp cho các thành phần thông qua dependency injection(DI).

Các thành phần của ứng dụng thường xác định nhiều view, được sắp sếp theo thứ bậc. Angular cung cấp dịch vụ bộ định tuyến (Router) giúp bạn có thể xác định đường dẫn điều hướng giữa các view. Các bộ định tuyến cung cấp khả năng điều hướng trong trình duyệt.



**Hình 2.3. Kiến trúc tổng quan của Angular**

* **Modules:**

Ứng dụng Angular sử dụng hệ thống module (NgModules. Một NgModules khai báo một bối cảnh biên dịch cho một tập hợp các thành phần dành riêng cho một miền ứng dụng, một quy trình làm việc hay tập hợp các khả năng liên quan chặt chẽ. Một NgModules có thể liên kết các thành phần của nó với mã nguồn liên quan chẳng hạn như các dịch vụ để tạo thành các đơn vị chức năng.

Mỗi ứng dụng Angular đều có một mô-đun gốc (AppModule) cung cấp cơ chế để khởi chạy ứng dụng. một ứng dụng thường có nhiều mô-đun chức năng.

Giống như các mô-đun JavaScript, NgModules có thể nhập các NgModules khác và cho phép chức năng của chính chúng được xuất và sử dụng bởi các NgModules khác.

Tổ chức mã nguồn của bạn thành các mô đun chức năng riêng biệt giúp quản lý các ứng dụng phức tạp và thiết kế để có thể sử dụng lại. Ngoài ra, kỹ thuật này cho phép bạn tải các mô-đun theo yêu cầu, để giảm thiểu số lượng mã cần được tải khi khởi động ứng dụng.

* **Components:**

Mỗi ứng dụng Angular có ít nhất một component, component gốc kết nối phân cấp component với mô hình đối tượng tài liệu trang (DOM). Mỗi thành phần định nghĩa một lớp chứa dữ liệu ứng dụng và logic và được liên kết với một mẫu HTML xác định một khung nhìn sẽ được hiển thị trong môi trường đích

* **Templates, directives, and data binding**

Một template kết hợp HTML với Angular markup để có thể chỉnh sửa các thành phần HTML trước khi chúng đc hiển thị. Các template directive cung cấp chương trình logic và đánh dấu ràng buộc kết nối dữ liệu ứng dụng và DOM. Có 2 loại ràng buộc dữ liệu:

* Event binding cho phép ứng dụng của bạn nhận phản hồi từ người dùng trong môi trường đích bằng cách cập nhật dữ liệu ứng dụng của bạn.
* Property binding cho phép bạn lấy các giá trị được tính toán từ dữ liệu ứng dụng của bạn vào HTML.

Trước khi một khung nhìn được hiển thị, Angular sẽ đánh giá các chỉ thị và xử lý các ràng buộc trong khuôn mẫu để sửa đổi các thành phần HTML và DOM theo dữ liệu và logic chương trình của bạn. Angular hỗ trợ liên kết dữ liệu hai chiều – những thay đối trong DOM cũng sẽ luôn được phản ánh trong dữ liệu chương trình của bạn.

* **Services and dependency injection**

Đối với dữ liệu hoặc logic không được liên kết với một khung cụ thể và bạn muốn chia sẻ giữa các component, ta có thể tạo một lớp service. Nó sẽ được đưa vào các component như một phần phụ thuộc.

Dependency injection (DI) cho phép bạn giữ cho các lớp thành phần của bạn gọn gàng và hiệu quả

* Routing

NgModule bộ định tuyến của Angular cung cấp một service cho phép bạn xác định đường dẫn điều hướng giữa các trạng thái ứng dụng khác nhau và khung nhìn phân cấp trong ứng dụng. Nó được mô phỏng theo các quy ước điều hướng trình duyệt quen thuộc.

### **2.2.3. Web Services**

#### **2.2.3.1. Giới thiệu về Web Services**

Web Services hay còn gọi là dịch vụ web là thành phần được xây dựng để có thể giao tiếp với các phần mềm, hệ thống khác qua môi trường Internet bằng cách sử dụng các công nghệ Internet tiêu chuẩn. Dịch vụ web được xác định với URL và không có giao diện người dùng mà chỉ giao tiếp với các thành phần khác thông qua các API (giao diện lâp trình ứng dụng). Nhờ sử dụng các công nghệ quen thuộc như XML, HTTP mà dịch vụ web c ó thể giao tiếp giữa các hệ thống mà sử dụng các ngôn ngữ và nền tảng khác nhau, đồng thời là một cách tiếp cận tuyệt vời để xây dựng các ứng dụng phân tán.

Web Services gồm ba thành phần cơ bản:

* SOAP (Simple Object Access Protocol): Một giao thức dựa trên XML để trao đổi thông điệp giữa các hệ thống thông qua HTTP. Đóng vai trò nền tảng cho các tương tác của Web Services.
* WSDL (Web Services Description Language): Ngôn ngữ dựa trên nền tảng XML dùng để định nghĩa giao diện của dịch vụ Web.
* UDDI (Universal Description, Discovery and Integration): Là một tiêu chuẩn dựa trên XML để mô tả, xuất bản và tìm kiếm các dịch vụ Web.

#### **2.2.3.2. RESTful Web Services**

REST là viết tắt của Representational State Transfer, là một tiêu chuẩn web dựa trên trên kiến trúc và sử dụng giao thức HTTP để trao đổi dữ liệu. Nó xoay quanh các tài nguyên trong đó mọi thành phần là tài nguyên và tài nguyên đó được truy cập bởi một giao diện chung sử dụng phương thức chuẩn HTTP. Trong kiến trúc REST, máy chủ REST chỉ đơn giản cung cấp quyền truy cập vào tài nguyên và các client truy cập và trình bày các tài nguyên đó. Mỗi tài nguyên được xác đinh bởi URI. REST sử dụng các cách biểu diễn khác nhau để biểu diễn tài nguyên như Text, JSON, XML.

Các phương thức HTTP sau được sử dụng phổ biến trong kiến trúc REST:

* GET: Cung cấp quyền truy cập chỉ đọc với tài nguyên
* POST: Được sử dụng để tạo tài nguyên mới
* PUT: Được sử dụng để cập nhật tài nguyên hiện có hoặc tạo tài nguyên mới
* DELETE: Được sử dụng để xóa tài nguyên
* OPTIONS: Được sử dụng để có được các hoạt động được hỗ trợ trên tài nguyên

RESTful Web Services là các dịch vụ web được xây dựng dựa trên kiến trúc REST. RESTful Web Services có dung lượng nhẹ, khả năng mở rộng và bảo trì tương đối dễ, thường được sử dụng để tạo các API cho nhiều ứng dụng ở các hệ thống khác nhau.

### **2.2.4. Laravel**

#### **2.2.4.1. Giới thiệu**

Laravel là một web Framework mã nguồn mở cho ngôn ngữ PHP, được tạo ra và phát triển bởi Taylor Otwell với phiên bản đầu tiên được ra mắt vào tháng 6 năm 2011. Được xây dựng với mục tiêu giúp cho quá trình xây dựng và phát triển các ứng dụng dễ dãng hơn. Đây là một frameword rất mạnh mẽ của PHP, được sự hỗ trợ, ưa chuộng và nhiều người sử dụng trong việc xây dựng các ứng dụng web cũng như các web service.

Hơn nữa Laravel được xây dựng theo mô hình MVC (Model-View-Controller), một mô hình phổ biến trong các web framework. Trong mô hình MVC, mã nguồn thì mã nguồn của ứng dụng được tổ chức thành ba thành phần chính, mỗi thành phần có nhiệm vụ riêng biêt:

* Model: Xác định cấu trúc dữ liệu và giao tiếp với cơ sở dữ liệu.
* Controller: Quản lý luồng dữ liệu và giao tiếp giữa Model và View.
* View: Chịu trách nhiệm giao tiếp với người dùng.

Nhờ vậy mà cấu trúc và cách tổ chức mã nguồn trong một dự án được sắp xếp một cách hợp lý dễ dàng cho việc bảo trì cũng như phát triển về lâu dài.

Laravel sử dụng Composer – một công cụ để quản lý các phụ thuộc trong PHP cho phép bạn khai báo các thư viện mà dự án của bạn đang sử dụng và sẽ quản lý (cài đặt, cập nhật). Với Composer bạn có thể thêm các gói và thư viện sẵn sàng của bên thứ ba vào trong dự án Laravel của bạn và quản lý chúng ở mọi nơi.

**\* Ưu điểm của Laravel:**

* Góp phần làm cho quá trình xây dựng và phát triển ứng dụng nhanh hơn. Các nhà phát triển chỉ cần tập trung vào sự phát triển của mình mà không phải lo lắng về một số nguyên tắc cơ bản chính khi phát triển ứng dụng
* Laravel sử dụng Eloquent ORM(Object-relational mapping) để làm việc với cơ sở dữ liệu, bạn không cần phải viết cac truy vẫn phức tạp để thực hiện các thao tác tạo, đọc, cập nhật và xóa.
* Với Laravel các nhà phát triển có thể dễ dàng mở rộng ứng dụng của mình.
* Qua các phiên bản, Laravel luôn được cập nhật liên tục với các cải tiến và bản vá để bảo vệ ứng dụng khỏi các mỗi đe dọa về bảo mât. Phiên bản hiện tại của Laravel là 5.7.
* Laravel có một cộng đồng người dùng và hỗ trợ vô cùng lớn, nhớ đó mà quá trình phát triển ứng dụng có thể dễ dàng hơn.

#### **2.2.4.2. Các tính năng của Laravel**

* **Truy cập cơ sở dữ liệu**
  + Laravel cung cấp một bộ công cụ tiên tiến để có thể giao tiếp với cơ sở dữ liệu. Với Database migrations cho phép bạn dễ dàng thiết kế và sửa đổi cơ sở dữ liệu theo cách độc lập với nền tảng. Đồng thời có thể được chạy với bất kỳ loại cơ sở dữ liệu nào mà Laravel hỗ trợ như: MySQL, PostgreQuery, MSSQL và SQLite. Với Query Builder cho phép bạn xây dựng và thực hiện các truy vấn trên mọi loại cơ sở dữ liệu khác nhau.
  + Với Eloquent Laravel cho phép bạn tương tác với cơ sở dữ liệu theo cách hướng đối tượng, đây là một tiêu chuẩn hiện đại. Với Eloquent chúng ta có thể tạo, truy xuất, cập nhật và xóa các bản ghi trên cơ sở dữ liệu mà không cần phải viết một dòng SQL (Structured Query Language). Ngoài ra, Eloquent cung cấp khả năng quản lý mối quan hệ mạnh mẽ và thậm chí nó có thể tự động xử lý phân trang cho ứng dụng của bạn.
* **Công cụ dòng lệnh**
* Laravel cũng cung cấp một công cụ giao diện dòng lệnh được gọi là Artisan. Với Artisan, các nhà phát triển có thể tương tác với ứng dụng để có thể kích hoạt các hành động như chạy database migrations, thực hiện kiểm thử đơn vị (unit test), và lên lịch một công việc. Hơn thế nữa với Artisan bạn hoàn toàn có thể chủ động mở rộng để thêm nhiều lệnh mới thực hiện bất kỳ chức năng nào mà bạn muốn.
* **Định tuyến**

Laravel cho phép nhà phát triển tổ chức tất cả các URL của ứng dụng thông qua các bộ định tuyến. Bạn có thể dễ dàng thêm mới, sửa đổi hay xóa chúng chỉ trong một tệp duy nhất.

* **View Template engine**

Trong Laravel Blade teample giúp nhà phát triển tách biệt các đoạn mã PHP với phần HTML trong các view, đồng thới có tích hợp thêm nhiều tính năng mạnh mẽ.

* **Lên kế hoạch cho công việc**

Từ phiên bản Laravel 5.0 thì đây là một bổ sung cho Artisan cho phép lập lịch trình cho các nhiệm vụ được thực hiện định kỳ do nhà phát triển tùy chỉnh.

* **Hệ thống tệp tin (File System)**

Được giới thiệu trong phiên bản Laravel 5.0, đây là lớp trừu tượng hệ thống tệp tin cho phép các hệ thống tệp tin cục bộ và dịch vụ lưu trữ dựa trên đám mây được cung cấp bởi Amazon S3 và Rackspace Cloud được sử dụng một cách minh bạch và theo cùng một cách.

* **Xác thực**

Laravel được xây dựng với các cơ chế xác thực cơ bản, bạn có thể cấu hình cho các trang web và API một cách dễ dàng chỉ với vài bước. Từ phiên bản 5.0 thì Laravel bao gồm gói tùy chọn cung cấp các cơ chế đơn giản hóa để xác thực với các nhà cung cấp bên thứ ba như Facebook, Twitter, Google, GitHub và Bitbucket.

### **2.2.5. Firebase**

Firebase là một nền tảng phát triển ứng dụng di động và ứng dụng web cung cấp cho các nhà phát triển rất nhiều công cụ và dịch vụ để giúp họ phát triển các ứng dụng chất lượng cao, phát triển cơ sở người dùng và kiếm thêm lợi nhuận. Cụ thể Firebase cung cấp các tính năng xoay quanh các dịch vụ đám mây cho phép người dùng lưu và truy xuất dữ liệu được truy cập từ bất kì thiết bị hay trình duyệt nào. Điều này có thể hữu ích cho những việc như nhắn tin trên đám mây, lưu trữ, báo cáo sự cố, thông báo, phân tích hay thậm chí tìm kiếm.

Điều nổi bật ở Firebase là Firebase sử dụng cơ sở dữ liệu thời gian thực và tất cả dữ liệu được tự động đồng bộ hóa trên tất cả người dùng - thay vì được lưu trữ trên điện thoại trước và sau đó gửi đi. Mặc dù điều này chắc chắn có lợi thế, nhưng điều đó có nghĩa là thiết bị người dùng luôn luôn cần kết nối internet để ứng dụng hoạt động. Đối với hầu hết các phần, đây là một cách cực kỳ mạnh mẽ và linh hoạt để xây dựng các ứng dụng hỗ trợ đám mây và là điều mà mọi nhà phát triển Android nên làm quen với nó.

Một số dịch vụ vủa Firebase:

* Xây dựng ứng dụng: Cloud Firestore, ML Kit, Cloud Funtions, Authentication, Hosting, Cloud Storage, Realtime Database.
* Cải thiện chất lượng của ứng dụng: Crashlytics, Performance Monitoring, Test Lab.
* Phát triển doanh nghiệp của bạn: In-App Messaging, Google Analytics, Predictions, A/B Testing, Cloud Messaging, Remote Config, Dynamic Links, App Indexing.

Tóm lại Firebase xử lý yếu tố trực tuyến cho phần backend của các ứng dụng của bạn, cho phép bạn tập trung vào giao diện người dùng và chức năng của ứng dụng. Tất cả điều này được thực hiện thông qua một SDK duy nhất với các API dễ sử dụng và tích hợp tuyệt vời vào Android Studio. Điều giúp bạn giảm bớt một số thao tác xử lý khi bạn làm việc kết hợp với phía máy chủ của riêng bạn bằng PHP và MySQL hoặc các thiết lập tương tự. Nó hoạt động với các ứng dụng Android, ứng dụng iOS và ứng dụng web và đặc biêt là nó miễn phí!

### **2.2.6. Google Maps Api**

Google Maps là một ứng dụng và công nghệ dịch vụ bản đồ trực tuyến dựa trên web được Google phát triển. Nó cung cấp những thông tin cụ thể chi tiết về các khu vực địa lý, địa điểm, bản đồ đường phố, đồng thời có thể lập các tuyến đường để chỉ đường cho người dùng đi bộ, xe máy hay ô tô của nhiều nơi trên nhiều quốc gia trên thế giới.

Ngoài ứng dụng Google Maps cho các ứng dụng di động hay ứng dụng web thì Google cũng phát triển các Google Maps Api cho các nhà phát triển ứng dụng để từ đó xây dựng và tùy chỉnh các bản đồ riêng cho ứng dụng của họ với nội dung và hình ảnh tùy chỉnh. Hiện tại Api của Google Maps có bốn loại bản đồ cơ bản: lộ trình, vệ tinh, kết hợp bản đồ ảnh hàng không với bản đồ đường, địa hình mà bạn có thể điều chỉnh bằng cách sử dụng các lớp và kiểu, điều khiển và sự kiện cũng như các dịch vụ và thư viện khác nhau.

Một số dịch vụ mà Google Maps cung cấp:

* Directions: Bạn có thể tính toán chỉ đường với nhiều loại phương thức vận chuyển bằng cách sử dụng DirectionsSevice. Khi có yêu cầu DirectionsSevice kết nối tới dịch vụ chỉ đường của Google Maps API để nhận yêu cầu chỉ đường và trả về đường dẫn tối ưu cho bên yêu cầu.
* Distance Matrix: Tính toán khoảng cách di chuyển và thời gian của hành trình giữa nhiều nguồn gốc và điểm đến bằng một phương thức di chuyển nhất định.
* Elevation: Cung cấp dữ liệu về độ cao cho các vị trí trên bề mặt trái đất, bao gồm cả các vị trí độ sâu dưới đại dương. Trong trường hợp Google không có các phép đo chính xác thì dịch vụ này sẽ nội suy và trả về giá trị trung bình của bốn vị trí gần nhất với vị trí đó. Ngoài ra bạn cũng có thể yêu cầu được lấy mẫu dọc theo các đường dẫn cho phép bạn tính toán các thay đổi độ cao tương đương dọc theo các tuyến đường đó.
* Geocoding: Quá trình chuyển đổi địa chỉ được gửi lên thành tọa độ địa lý cụ thể (kinh độ và vĩ độ) và ngược lại, đồng thời bạn có thể sử dụng để đặt điểm đánh dấu hay định vị bản đồ. Bạn cũng có thể sử dụng để tìm địa chỉ cho một ID của địa điểm cụ thể.
* Maximum Zoom Imagery: Google Maps API cung cấp bản đồ với các mức thu phóng khác nhau cho hình ảnh bản đồ (các mức từ 0 đến 18). Do hình ảnh vệ tinh không phải lúc nào cũng có sẵn ở các mức thu phóng cao cho các địa điểm ở xa như khu vực dân cư thưa thớt hay vùng đại dương mở vậy bạn có thể muốn biết mức thu phóng cao nhất cho hình ảnh tại một vị trí nhất định thì MaxZoomService giúp cung cấp một giao diện đơn giản để khám phá mức thu phóng tối đa tại vị trí nhất định mà Google Maps có hình ảnh vệ tinh.
* Street View: Cung cấp những hình ảnh 360 độ toàn cảnh từ các đường được chỉ định trong khu vực phủ sóng.

## 2.3. Xác định yêu cầu

### **2.3.1. Yêu cầu của ứng dụng**

Ứng dụng hướng tới hỗ trợ người sử dụng là các chủ phương tiện muốn chia sẻ ghế trống của mình và hàng khách muốn tìm một chuyến đi cho mình, cần đảm bảo một số chức năng cơ bản sau:

* Các chủ phương tiện có thể tạo một chuyến đi với thông tin cụ thể về điểm đi, điểm đến, thời gian, và các yêu cầu cụ thể rồi đăng một cách công khai lên cộng đồng.
* Các chủ phương tiện có thể cung cấp thông tin về phương tiện của mình để cho phía hàng khách có thể quan tâm
* Các hàng khách có thể tìm kiếm và chọn một chuyến đi phù hợp cho mình với địa điểm và thời gian được chọn.
* Sau khi hoàn thánh chuyến đi, hàng khách có thể tiến hành nhận xét, đánh giá ý kiến đối với chủ phương tiện
* Hàng khách có thể trò chuyện trực tiếp với chủ phương tiện thông qua ứng dụng.
* Tất cả người dùng có thể quản lý thông tin cá nhân của mình trên ứng dụng.

### **2.3.2. Đặc tả yêu cầu**

Ứng dụng xây dựng yêu cầu hỗ trợ các đối tượng là chủ phương tiện muốn chia sẻ ghế trống trên phương tiện của mình khi muốn đi đâu đó và các hàng khách muốn tìm một chuyến đi cho mình và muốn có người đồng hành cùng.

Cụ thể, chủ phương tiện khi muốn đi đâu đó với phương tiện của mình mà muốn chia sẻ ghế trống trên xe với mọi người thì chủ phương tiện có thể tiến hành chia sẻ thông tin chuyến đi của mình lên hệ thống để tất cả mọi người quan tâm có thể biết đến. Chủ phương tiện có thể quản lý thông tin của mình về thông tin cá nhân, phương tiện của mình để tạo sự tin tưởng với những người quan tâm.

Các hàng khách khi có nhu cầu di chuyển đi đâu đó có thể tìm kiếm những chuyến đi phù hợp với bản thân và sau đó có thể tiến hành đặt một chuyến xe cho mình, hàng khách cũng có thể liên lạc với chủ phương tiện để trao đổi những thông tin cần thiết.

Tất cả người dùng có thể quản lý thông tin cá nhân của mình và chia sẻ công khai thông tin đó để tạo sự tin tưởng với những người dùng khác quan tâm.

### **2.3.2. Yêu cầu cụ thể**

#### **2.3.2.1. Yêu cầu chức năng**

|  |  |  |
| --- | --- | --- |
| **STT** | **Yêu cầu chức năng** | **Mô tả** |
| 1 | Tạo chuyến đi | Chức năng cho phép chủ phương tiện có thể tạo một chuyến đi với thông tin đầy đủ về điểm đi, điểm đến, thời gian đi, chi phí và mô tả về chuyến đi sau đó chuyến đi sẽ được công khai lên hệ thống. |
| 2 | Tìm kiếm chuyến đi | Chức năng cho phép các hàng khách – những người có nhu cầu tìm một chuyến đi phù hợp với mình có thể tìm kiếm các chuyến đi mà các chủ phương tiện đã công khai lên hệ thống. |
| 3 | Đặt chuyến đi | Chức năng cho phép hàng khách sau khi chọn được chuyến đi phù hợp có thể đặt trước với chủ xe và thông báo tới chủ xe |
| 4 | Quản lý phương tiện | Chức năng cho phép chủ phương tiện có thể thêm thông tin về phương tiện của mình đồng thời cũng có thể sửa, xóa phương tiện đã thêm. |
| 5 | Quản lý thông tin cá nhân | Chức năng cho phép người dùng có thể quản lý thông tin cá nhân của mình trên hệ thống, có thể tùy chỉnh thông tin này để công khai lên hệ thống. |
| 6 | Chat | Chức năng cho phép hàng khách và chú phương tiện có thể trao đổi tin nhắn với nhau qua hệ thống để tra đổi thông tin về chuyến đi. |

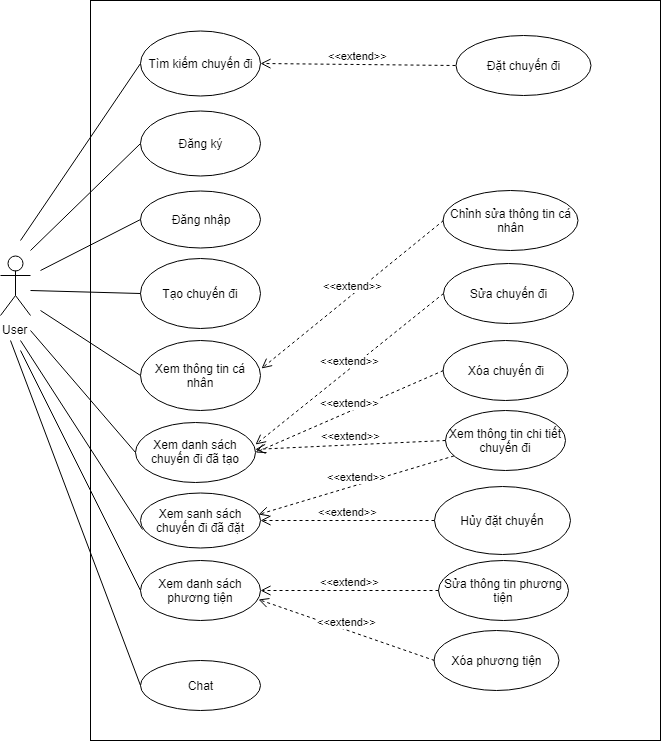
**Bảng 2.1. Yêu cầu chức năng của ứng dụng**

**2.3.2.2. Yêu cầu phi chức năng**

|  |  |  |
| --- | --- | --- |
| **STT** | **Yêu cầu phi chức năng** | **Mô tả** |
| 1 | Yêu cầu về hiệu năng | Hệ thống phản hồi nhanh, thời gian trả về kết quả tìm kiếm không quá 3s khi người dùng thực hiện yêu cầu. |
| 2 | Yêu cầu về giao diện | Giao diện dễ sử dụng, thao tác, có thể thành thạo sau một số lần sử dụng |
| 3 | Dễ cài đặt | Ứng dụng dễ cài đặt trên nhiều thiết bị android |
| 4 | Tương thích | Ứng dụng hỗ trợ các thiết bị có Android 4+ |
| 5 | An toàn thông tin | Thông tin của người dùng phải được đảm bảo an toàn bảo mật, cần xác thực khi sử dụng ứng dụng. |

**Bảng 2.2. Yêu cầu phi chức năng của ứng dụng**

## 2.3. Mô hình ca sử dụng



**Hình 2.4. Mô hình ca sử dụng**

**2.3.1. Đăng ký**

**Mô tả ngắn gọn**

Người dùng đăng ký tài khoản để sử dụng hệ thống

**Luồng sự kiện**

Trường hợp này sử dụng khi cần đăng ký tài khoản để vào sử dụng ứng dụng

* **Luồng cơ bản**

Ứng dụng hiển thị khung cho phép nhập thông tin như firstname, lastname, email, password. Nếu thông tin được nhập đầy đủ và hợp lệ thì ứng dụng sẽ đưa ra thông báo đăng ký thành công và chuyển đến màn hình đăng nhập.

* **Luồng thay thế**

1. Nếu các thông tin cần nhập bị trống, người dùng không thể đăng ký tài khoản sử dụng đồng thời ứng dụng đưa ra thông báo yêu cầu người dùng nhập đầy đủ.
2. Nếu các thông tin cần nhập không hợp lệ như email hợp lệ, password cần ít nhất 6 ký tự thì người dùng cũng không thể đăng ký tài khoản đồng thời ứng dụng đưa ra cảnh báo yêu cầu người dùng cần nhập lại.

**Yêu cầu đặc biệt**

Không

**Điều kiện đầu**

Không

**Điều kiện sau**

Sau khi đăng ký thành công người dùng được chuyển đến màn hình đăng nhập.

**Ngoại lệ**

Không

**2.3.2. Đăng nhập**

**Mô tả ngắn gọn**

Người dùng đăng nhập vào sử dụng ứng dụng

**Luồng sự kiện**

Trường hợp này sử dụng khi bắt đầu vào sử dụng ứng dụng

* **Luồng cơ bản**

Ứng dụng hiển thị khung cho phép nhập thông tin email, password. Nếu thông tin được nhập đầy đủ và hợp lệ thì ứng dụng sẽ đưa ra thông báo đăng nhập thành công.

* **Luồng thay thế**

1. Nếu các thông tin cần nhập bị trống, người dùng không thể đăng nhập vào ứng dụng đồng thời ứng dụng đưa ra thông báo yêu cầu người dùng nhập đầy đủ.
2. Nếu các thông tin cần nhập email, password bị sai, người dùng không thể đăng nhập vào ứng dụng. Đồng thời đưa ra thông báo yêu cầu người dùng kiểm tra và nhập lại.

**Yêu cầu đặc biệt**

Không.

**Điều kiện đầu**

Người dùng đã có tài khoản trong hệ thống.

**Điều kiện sau**

Sau khi đăng nhập người dùng được chuyển đến màn hình trang chủ của ứng dụng.

**Ngoại lệ**

Không.

**2.3.3. Tìm kiếm chuyến đi**

**Mô tả ngắn gọn**

Hàng khách muốn tìm kiếm một chuyến đi

**Luồng sự kiện**

* Luồng cơ bản

Sau khi người dùng đăng nhập và vào màn hình trang chủ của ứng dụng, người dùng chọn vào menu tìm kiếm hoặc chọn vào nút tìm kiếm chuyến đi trên trang chủ, ứng dụng sẽ đưa người dùng đến màn hình tìm kiếm. Tại màn hình này yêu cầu người dùng phải nhập các thông tin về điểm đi, điểm đến và thời gian cho chuyến đi. Sau khi nhập ứng dụng sẽ tiến hành tìm kiếm các chuyến đi theo các điều kiện đã nhập và hiện ra một danh sách các chuyến đi phù hợp.

* Luồng thay thế

Nếu không tìm được chuyến đi nào, thì màn hình hiện danh sách sẽ trống.

**Yêu cầu đặc biệt**

Không

**Điều kiện đầu**

Người dùng phải đăng nhập vào ứng dụng

**Điều kiện sau**

Không

**Ngoại lệ**

Không

**2.3.4. Xem chi tiết chuyến đi**

**Mô tả ngắn gọn**

Người dùng xem chi tiết thông tin chuyến đi

**Luồng sự kiện**

* Luồng cơ bản

Khi người dùng chọn một chuyến đi trong danh sách, ứng dụng sẽ chuyển đến màn hình hiển thị thông tin chi tiết của chuyến đi

* Luồng thay thế

Không

**Yêu cầu đặc biệt**

Không

**Điều kiện đầu**

Người dùng phải đăng nhập vào ứng dụng

**Điều kiện sau**

Không

**Ngoại lệ**

Không

**2.3.5. Đặt chuyến đi**

**Mô tả ngắn gọn**

Hàng khách sau khi chọn được chuyến đi phù hợp muốn đặt chỗ với chủ phương tiện

**Luồng sự kiện**

* Luồng cơ bản

Trong khi xem chi tiết thông tin của chuyến đi, hàng khách có thể chọn vào nút đặt chuyến, ứng dụng sẽ đưa ra thông báo xác nhận hàng khách có chắc chắn muốn đặt chuyến này hay không. Nếu chọn đồng ý thì ứng dụng sẽ hiện ra thông báo đã đặt chuyến thành công, đồng thời gửi thông báo đến phía chủ phương tiện biết có người đã đặt chuyến đi đó.

* Luồng thay thế

Nếu chuyến đi không được chú phương tiện đặt là trạng thái là tự động chấp nhận thì một thông báo yêu cầu xác nhận sẽ được gửi tới chủ phương tiện.

Nếu chủ phương tiện xác nhận đồng ý chấp nhận thì chuyến đi đó sẽ được chuyển sang trạng thái được đặt.

Nếu chủ phương tiện hủy thì chuyến đi đó sẽ không được đặt và chuyển sang trạng thái vừa tạo.

**Yêu cầu đặc biệt**

Không

**Điều kiện đầu**

Người dùng phải đăng nhập vào ứng dụng

**Điều kiện sau**

Không

**Ngoại lệ**

Không

**2.3.6. Tạo chuyến đi**

**Mô tả ngắn gọn**

Tài xế muốn đăng thông tin để chia sẻ công khai chuyến đi của mình

**Luồng sự kiện**

* Luồng cơ bản

Sau khi đăng nhập, người dùng chọn vào mục menu tạo chuyến đi mới, ứng dụng sẽ đưa tài xế đến một loạt các màn hình để nhập các thông tin về điểm đi, điểm đến, ngày đi, giờ đi, chi phi mong muốn được chia sẻ, tự động chấp nhận khi có yêu cầu và mô tả thông tin mong muốn về chuyến đi đó. Khi chọn nút đăng chuyến đi ứng dụng sẽ hiện thông báo xem ý kiến của người dùng có chắc chắn muốn đăng chuyến đi này hay không.

* Luồng thay thế

Khi người dùng nhập thông tin không hợp lệ ứng dụng sẽ hiện ra thông báo yêu cầu người dùng kiểm tra lại thông tin đã nhập.

**Yêu cầu đặc biệt**

Không

**Điều kiện đầu**

Người dùng phải đăng nhập vào hệ thống

**Điều kiện sau**

Không

**Ngoại lệ**

Không

**2.3.7. Xem thông tin cá nhân**

**Mô tả ngắn gọn**

Ca sử dụng cho phép người dùng có thể xem thông tin cá nhân chi tiết của mình

**Luồng sự kiện**

* Luồng cơ bản

Ứng dụng hiển thị thanh menu các chức năng

Người dùng chọn vào menu “Giới thiệu”, ứng dụng sẽ chuyển đến trang hiện thông tin chi tiết cá nhân mà người dùng đã cung cấp trên hệ thống.

* Luồng thay thế

Không

**Yêu cầu đặc biệt**

Không

**Điều kiện đầu**

Người dùng phải đăng nhập vào hệ thống

**Điều kiện sau**

Không

**Ngoại lệ**

Không

**2.3.8. Sửa thông tin cá nhân**

**Mô tả ngắn gọn**

Ca sử dụng cho phép người dùng có thể truy cập để chỉnh sửa những thông tin cá nhân cần thiết của mình

**Luồng sự kiện**

* Luồng cơ bản

Ca sử dụng bắt đầu khi từ ca sử dụng xem thông tin cá nhân của mình, người dùng chọn vào nút “Sửa”.

Ứng dụng sẽ chuyển sang trang cho phép người dùng sửa những thông tin cá nhân hiện có đồng thời thêm những thông tin mới.

Sau khi nhập xong chọn nút “Sửa đổi”, nếu các thông tin hợp lệ ứng dụng sẽ đưa ra thông báo cho người dùng là đã sửa thành công.

* Luồng thay thế

Nếu người dùng chọn nút “quay lại” trên màn hình thì ứng dụng sẽ quay lại màn hình thông tin cá nhân chi tiết mà không thực hiện sửa đổi.

Nếu các thông tin nhập không hợp lệ ứng dụng sẽ đưa ra thông báo yêu cầu người dùng kiểm tra lại.

**Yêu cầu đặc biệt**

Không

**Điều kiện đầu**

Người dùng phải đăng nhập vào ứng dụng

**Điều kiện sau**

Nếu sửa thành công ứng dụng sẽ chuyển đến trang thông tin cá nhân với thông tin đã được sửa đổi.

**Ngoại lệ**

Không

**2.3.9. Xem danh sách chuyến đi đã tạo**

**Mô tả ngắn gọn**

Ca sử dụng cho phép những tài xế có thể xem lại danh sách những chuyến đi mà mình đã tạo được đăng lên hệ thống và trạng thái của những chuyến đi đó.

**Luồng sự kiện**

* Luồng cơ bản

Ca sử dụng bắt đầu khi người dùng chọn chức năng xem lịch sử trên thanh menu.

Ứng dụng sẽ chuyển đến màn hình “Chuyến đi của bạn”, khi chọn tab “Chuyến đi đã tạo” màn hình sẽ hiển thị danh sách các chuyến đi đã tạo, các danh sách này được chia thành 2 danh sách gồm những chuyến đi hiện tại vừa mới tạo và những chuyến đi đã đi trong quá khứ.

Với những chuyến đi người dùng vừa mới tạo thì sẽ hiện menu “Sửa chuyến đi” và “Xóa chuyến đi”.

* Luồng thay thế

Không

**Yêu cầu đặc biệt**

Không

**Điều kiện đầu**

Người dùng phải đăng nhập vào ứng dụng

**Điều kiện sau**

Không

**Ngoại lệ**

Không

**2.3.10. Xem danh sách chuyến đi đã đặt**

**Mô tả ngắn gọn**

Ca sử dụng cho phép những tài xế có thể xem lại danh sách những chuyến đi mà mình đã đặt trên hệ thống và trạng thái của những chuyến đi đó.

**Luồng sự kiện**

* Luồng cơ bản

Ca sử dụng bắt đầu khi ở màn hình “Chuyến đi của bạn” người dùng chọn tab “Chuyến đi đã đặt”

Màn hình sẽ chuyển sang hiển thị danh sách các chuyến đi mà người dùng đã đặt được chia thành 2 danh sách gồm những chuyến đi đang đăt hiện tại và những chuyến đi mà người dùng đã đi trong quá khứ.

Với những chuyến đi người dùng đang đặt thì sẽ có một menu “Hủy chuyến” cho phép người dùng hủy chuyến.

* Luồng thay thế

Không

**Yêu cầu đặc biệt**

Không

**Điều kiện đầu**

Người dùng phải đăng nhập vào hệ thống

**Điều kiện sau**

Không

**Ngoại lệ**

Không

**2.3.11. Xem chi tiết chuyến đi**

**Mô tả ngắn gọn**

Ca sử dụng cho phép người dùng có thể xem thông tin chi tiết của một chuyến đi nào đó

**Luồng sự kiện**

* Luồng cơ bản

Ca sử dụng bắt đầu khi người dùng chọn vào một chuyến đi bất kì từ danh sách

Ứng dụng chuyển sang màn hình hiển thị thông tin chi tiết của chuyến đi

* Luồng thay thế

**Yêu cầu đặc biệt**

Không

**Điều kiện đầu**

Người dùng phải đăng nhập vào hệ thống

**Điều kiện sau**

Không

**Ngoại lệ**

Không

**2.3.12. Sửa chuyến đi**

**Mô tả ngắn gọn**

Ca sử dụng cho phép tài xế có thể chỉnh sửa những thông tin mà mình vừa mới tạo

**Luồng sự kiện**

* Luồng cơ bản

Từ màn hình hiển thị danh sách các chuyến đi đã tạo, với những chuyến đi mà tài xế vừa mới tạo, tài xế chọn vào menu “Sửa”.

Ứng dụng chuyển sang màn hình cho phép tài xế nhập các thông tin để chỉnh sửa những thông tin cần thiết.

Khi người dùng chọn nút “Sửa”, nếu các thông tin vừa nhập là hợp lệ thì ứng dụng sẽ đưa ra thông báo đã sửa đổi thành công.

* Luồng thay thế

Nếu các thông tin nhập chưa hợp lệ ứng dụng sẽ đưa ra thông báo yêu cầu người dùng kiểm tra lại các mục đã nhập.

Nếu người dùng chọn nút “Quay lại” trên màn hình, ứng dụng sẽ quay lại màn hình hiển thị danh sách chuyến đi trước đó với các thông tin như cũ.

**Yêu cầu đặc biệt**

Tài xế chỉ có thể sửa những chuyến đi vừa tạo hoặc có người đang đặt, không thể sửa được những chuyến đi đã đi trong quá khứ

**Điều kiện đầu**

Người dùng phải đăng nhập vào hệ thống

**Điều kiện sau**

Sau khi sửa thành công, ứng dụng sẽ chuyển sang màn hình hiển thị danh sách các chuyến đi đã tạo với thông tin đã được cập nhật.

Với những chuyến đi đang có người đặt thì ứng dụng sẽ gửi thông báo qua ứng dụng tới hàng khách đang đặt chuyến rằng thông tin chuyến đi đã được chỉnh sửa.

**Ngoại lệ**

Không

**2.3.13. Xóa chuyến đi**

**Mô tả ngắn gọn**

Ca sử dụng cho phép tài xế có thể xóa chuyến đi của mình vừa tạo

**Luồng sự kiện**

* Luồng cơ bản

Từ màn hình hiển thị danh sách các chuyến đi đã tạo, với những chuyến đi mà tài xế vừa mới tạo, tài xế chọn vào menu “Xóa”.

Ứng dụng sẽ đưa ra thông báo yêu cầu tài xế xác nhận xem có chắc chắc muốn xóa chuyến đi này không.

Nếu tài xế xác nhận có thì sẽ chuyến đi đó sẽ bị xóa trên hệ thống và thông báo xóa thành công tới tài xế.

* Luồng thay thế

Không

**Yêu cầu đặc biệt**

Tài xế chỉ có thể xóa những chuyến đi vừa tạo hoặc có người đang đặt, không thể xóa được những chuyến đi đã đi trong quá khứ

**Điều kiện đầu**

Người dùng đã đăng nhập vào hệ thống.

**Điều kiện sau**

Sau khi xóa thành công, màn hình danh sách chuyến đi đã tạo sẽ được cập nhật lại thông tin mới.

Với những chuyến đi đang có người đặt thì ứng dụng sẽ gửi thông báo qua ứng dụng tới hàng khách đang đặt chuyến rằng chuyến đi đã bị xóa.

**Ngoại lệ**

Không

**2.3.14. Hủy chuyến đi**

**Mô tả ngắn gọn**

Ca sử dụng cho phép người dùng là hàng khách có thể hủy đặt chuyến với chuyến đi mà mình vừa mới đặt mà chưa đi.

**Luồng sự kiện**

* Luồng cơ bản

Từ màn hình hiển thị danh sách các chuyến đi đã đặt, với những chuyến đi mà người dùng vừa đặt mà chưa đi, chọn vào menu “Hủy chuyến”.

Ứng dụng sẽ đưa ra thông báo yêu cầu người dùng xác nhận xem có chắc chắc muốn hủy chuyến đi này không.

Nếu tài xế người dùng có thì sẽ chuyến đi đó sẽ bị hủy trên hệ thống và thông báo xóa thành công tới người dùng đồng thời gửi thông báo tới tài xế qua ứng dụng.

* Luồng thay thế

Không

**Yêu cầu đặc biệt**

Không

**Điều kiện đầu**

Người dùng phải đăng nhập vào ứng dụng.

**Điều kiện sau**

Sau khi hủy thành công ứng dụng sẽ gửi thông báo qua ứng dụng tới tài xế để thông báo rằng hàng khách đã hủy chuyến

**Ngoại lệ**

Không

**2.3.15. Xem danh sách phương tiện**

**Mô tả ngắn gọn**

Ca sử dụng cho phép người dùng xem danh sách các phương tiện của mình đã thêm trên hệ thống.

**Luồng sự kiện**

* Luồng cơ bản

Từ danh sách menu của ứng dụng người dùng chọn mục “Phương tiện của bạn”, ứng dụng sẽ chuyển sang màn hình hiển thị danh sách các phương tiện của người dùng.

* Luồng thay thế

Không

**Yêu cầu đặc biệt**

Không.

**Điều kiện đầu**

Người dùng phải đăng nhập vào hệ thống.

**Điều kiện sau**

Không.

**Ngoại lệ**

Không.

**2.3.16. Thêm phương tiện**

**Mô tả ngắn gọn**

Ca sử dụng cho phép người dùng thêm thông tin về phương tiện lên hệ thống

**Luồng sự kiện**

* Luồng cơ bản

Tại màn hình danh sách các phương tiện, người dùng chọn vào nút “Thêm phương tiện” ứng dụng sẽ chuyển sang màn cho phép người dùng nhập một loạt các thông tin về phương tiện như: ảnh, biển số xe, mô tả,…

Người dùng chọn nút “Thêm”, nếu các thông tin vừa nhập là hợp lệ ứng dụng sẽ đưa ra thông báo cho người dùng là đã thêm thành công rồi chuyển sang màn hình danh sách phương tiện

* Luồng thay thế

Nếu các thông tin vừa nhập không hợp lệ ứng dụng đưa ra thông báo yêu cầu người dùng kiểm tra lại các thông tin đã nhập.

**Yêu cầu đặc biệt**

Không

**Điều kiện đầu**

Người dùng phải đăng nhập vào ứng dụng

**Điều kiện sau**

Sau khi thêm thành công, ứng dụng chuyển sang màn hình danh sách các phương tiện.

**Ngoại lệ**

Không

**2.3.17. Sửa phương tiện**

**Mô tả ngắn gọn**

Ca sử dụng cho phép người dùng sửa thông tin về phương tiện hiện có

**Luồng sự kiện**

* Luồng cơ bản

Tại màn hình danh sách các phương tiện, người dùng chọn vào menu “Sửa” được gắn với từng phương tiện ứng dụng sẽ chuyển sang màn cho phép người dùng sửa một loạt các thông tin về phương tiện như: ảnh, biển số xe, mô tả,…

Người dùng chọn nút “Sửa”, nếu các thông tin vừa nhập là hợp lệ ứng dụng sẽ đưa ra thông báo cho người dùng là đã sửa thành công rồi chuyển sang màn hình danh sách phương tiện

* Luồng thay thế

Nếu các thông tin vừa nhập không hợp lệ ứng dụng đưa ra thông báo yêu cầu người dùng kiểm tra lại các thông tin đã nhập.

**Yêu cầu đặc biệt**

Không

**Điều kiện đầu**

Người dùng phải đăng nhập vào ứng dụng

**Điều kiện sau**

Sau khi sửa thành công, ứng dụng chuyển sang màn hình danh sách các phương tiện.

**Ngoại lệ**

Không

**2.3.18. Xóa phương tiện**

**Mô tả ngắn gọn**

Ca sử dụng cho phép người dùng xóa một phương tiện hiện có

**Luồng sự kiện**

* Luồng cơ bản

Tại màn hình danh sách các phương tiện, người dùng chọn vào menu “Xóa” được gắn với từng phương tiện ứng dụng sẽ đưa thông báo yêu cầu người dùng xác nhận có chắc chắc muốn xóa phương tiện này hay không.

Nếu người dùng chọn có thì phương tiện này sẽ được xóa và danh sách phương tiện sẽ được cập nhật lại theo thông tin mới.

* Luồng thay thế

Không

**Yêu cầu đặc biệt**

Không

**Điều kiện đầu**

Người dùng phải đăng nhập vào ứng dụng

**Điều kiện sau**

Không

**Ngoại lệ**

Không

**2.3.19. Chat**

**Mô tả ngắn gọn**

Ca sử dụng cho phép tài xế và hàng khách có thể gửi tin nhắn để trao đổi trực tiếp với nhau

**Luồng sự kiện**

* Luồng cơ bản

Ca sử dụng bắt đầu khi hàng khách chọn mục “tin nhắn” khi vào trang thông tin cá nhân của người khác khi thực hiện tìm kiếm chuyến đi.

Hàng khách và tài xế có thể gửi tin nhắn trực tiếp với nhau.

* Luồng thay thế

Không

**Yêu cầu đặc biệt**

Không

**Điều kiện đầu**

Người dùng phải đăng nhập vào hệ thống

**Điều kiện sau**

Không

**Ngoại lệ**

Không

**2.4. Cơ sở dữ liệu**

Cơ sở dữ liệu được lưu thành 2 phần. Một phần được lưu trữ trên firebase với mục đích lưu thông tin người dùng, các file người dùng đã tải lên, nội dung chat giữa những người dùng với nhau. Phần khác được lưu trữ trên server sử dụng hệ quản trị cơ sở dữ liệu MySQL để lưu những thông tin khác liên quan đến chuyến đi của người dùng.

**CHƯƠNG 3. KẾT QUẢ**

**3.1. Firebase với việc phát triển một số chức năng cho ứng dụng**

Để sử dụng được các dịch vụ của Firebase đầu tiên ta cần đăng ký tài khoản trên Firebase sau đó mới có thể tiến hành truy cập và sử dụng các dịch vụ mà Firebase hỗ trợ.

* Tạo project và liên kết tới các dịch vụ của firebase
* Đăng nhập tới bảng điều khiển của firebase.
* Chọn “Add a project” rồi nhập các thông tin để tiến hành thêm một project mới.
* Sau khi tạo xong ta tiến hành thêm Firebase vào ứng dụng android bằng cách nhập các thông tin về package name, App nickname, certificate SHA-1. Sau khi hoàn thành ta sẽ nhận được 1 file cấu hình google-services.json.
* Thêm file cấu hình trên vào project Nativescript vào đường dẫn “app/App\_Resources/Android/”.
* Tiến hành chạy lệnh “tns plugin add nativescript-plugin-firebase” để thêm plugin “NativeScript Firebase plugin” vào project giúp hỗ trợ kết nối ứng dụng với Firebase.
* Khởi tạo kết nối đên firebase bằng đoạn mã sau:

const firebase = require("nativescript-plugin-firebase");

@Component({

// ...

})

export class AppComponent implements OnInit {

ngOnInit() {

firebase.init({

// Optionally pass in properties for database, authentication and cloud messaging,

// see their respective docs.

}).then(

() => {

console.log("firebase.init done");

},

error => {

console.log(`firebase.init error: ${error}`);

}

);

}

}

**3.1.1. Xác thực với Authentication**

Thông qua dịch vụ Authentication của firebase giúp ta xây dựng chức năng xác thực người dùng của ứng dụng để cung cấp tài khoản cho người dùng của ứng dụng và xác thực mỗi khi người dùng vào sử dụng ứng dụng qua email, tài khoản google, facebook:

* Trong bảng điều khiển của Firebase, chọn Authentication
* Tiếp đến chọn mục “Sign-in method” và kích hoạt các phương thức xác thực qua Email/Password, Google, Facebook.

Sau khi hoàn thành ta có thể xây dựng các chức năng xác thực với Email/password, Google và Facebook

* Đăng ký với Email/password: Người dùng có thể dễ dàng đăng ký tài khoản với email cá nhân và tạo password cho riêng mình

firebase.createUser({

email: 'eddyverbruggen@gmail.com',

password: 'firebase'

}).then(

function (user) {

dialogs.alert({

title: "User created",

message: "email: " + user.email,

okButtonText: "Nice!"

})

},

function (errorMessage) {

dialogs.alert({

title: "No user created",

message: errorMessage,

okButtonText: "OK, got it"

})

}

);

* Đăng ký bằng tài khoản Google, Facebook: Người dùng có thể sử dụng tài khoản của Google hay Facebook để đăng ký và sử dụng hệ thống.

Với Google:

firebase.login({

type: firebase.LoginType.GOOGLE,

// Optional

googleOptions: {

hostedDomain: "mygsuitedomain.com"

}

}).then(

function (result) {

JSON.stringify(result);

},

function (errorMessage) {

console.log(errorMessage);

}

);

Với Facebook ta cần tạo project dành riêng trong chế độ cho nhà phát triển của Facebook và cung cấp App-ID và App-Secret khi kích hoạt dịch vụ xác thực qua facebook

firebase.login({

type: firebase.LoginType.FACEBOOK,

// Optional

facebookOptions: {

// defaults to ['public\_profile', 'email']

scope: ['public\_profile', 'email']

}

}).then(

function (result) {

JSON.stringify(result);

},

function (errorMessage) {

console.log(errorMessage);

}

);

**3.1.2. Realtime Database cho chức năng chat trực tuyến giữa khách hàng và chủ phương tiện**

Với dịch vụ Realtime database của Firebase là một cơ sở dữ liệu NoSQL được lưu và đồng bộ trên đám mây theo thời gian thực với mỗi client được kết nối và vấn khả dụng khi offline. Mỗi khi dữ liệu có sự thay đổi, các thiết bị được kết nối sẽ nhận ngay được những thay đổi đó. Chính vì vây chúng ta sử dụng dịch vụ này của Firebase để xây dựng chức năng chat trực tuyến cho ứng dụng, mỗi khi có tin nhắn mới dữ liệu tin nhắn ngay lập tức sẽ được cập nhật vào cơ sở dữ liệu này và được gửi đến người nhận.

Trước hết ta cần kích hoạt module “realtimedb” khi tiến hành thêm plugin “NativeScript Firebase plugin” vào project.

* Để dữ liệu vẫn được lưu trên thiết bị khi thiết bị ngoại tuyến ta cần cài đặt thuộc tính “persist” là “true” trong hàm khởi tạo của Firebase. Lúc này mọi tin nhắn đã gửi và nhận sẽ luôn được lưu ngoại tuyến trên thiết bị và luôn sẵn sàng để xem.
* Để đẩy dữ liệu lên Firebase ta sử dụng mã lệnh sau:

firebase.push(

'/Chats/' + id,

{

'message': message,

'to': id,

'from': BackendService.userId,

'date': 0 – Date.now()

}

).then(

function (result:any) {

return 'chatted';

},

function (errorMessage:any) {

console.log("created key: " + result.key);

}

);

3.2. Xây dựng các RESTful API cho ứng dụng trên Laravel

Ta có thể sử dụng Framework Laravel để xây dựng các API cho ứng dụng để nhận các yêu cầu từ phía client để tiến hành xử lý và trả về kết quả cho client với dữ liệu dưới dạng JSON

Sau đây là danh sách các API đã được xây dựng:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Method** |  | **Mô tả** |
| 1 | POST | api/profile/checkUser | Kiểm tra thông tin người dùng khi đăng nhập |
| 2 | GET | api/profile/getProfile  /{userId} | Lấy thông tin người dùng |
| 3 | PUT | api/profile/updateProfile  /{userId} | Chỉnh sửa thông tin người dùng |
| 4 | GET | api/profile/getUserToken  /{userId} | Lấy token của người dùng |
| 5 | POST | api/vehicle/createVehicleInfo | Tạo mới phương tiện |
| 6 | GET | api/vehicle/getVehicleInfos  /{userId} | Lấy danh sách phương tiện của người dùng |
| 7 | PUT | api/vehicle/editVehicleInfo  /{vehicleId} | Chỉnh sửa thông tin phương tiện |
| 8 | DELETE | api/vehicle/deleteVehicleInfo  /{vehicleId} | Xóa phương tiện |
| 9 | GET | api/vehicle/getVehicleInfo  /{vehicleId} | Lấy thông tin chi tiết của một phương tiện |
| 10 | POST | api/ride/createNewRide | Tạo chuyến đi mới |
| 11 | PUT | api/ride/editRide  /{rideId} | Chỉnh sửa một chuyến đi |
| 12 | GET | api/ride/searchRide  /{uid}/{latFrom}/{longForm}  /{latTo}/{longTo}/{date}  /{date}/{time} | Tìm kiếm chuyến đi |
| 13 | GET | api/ride/getRideDetail/{rideId} | Lấy thông tin chi tiết của một chuyến đi |
| 14 | PUT | api/ride/bookRide  /{rideId}/{customerId} | Đặt chuyến đi |
| 15 | PUT | api/ride/bookRideNotAutoAccept  /{rideId}/{customerId} | Đặt chuyến đi với những chuyến đi không đặt trạng thái tự động chấp nhận khi có yêu cầu |
| 16 | PUT | api/ride/cancelRequireBookRide  /{rideId} | Hủy yêu cầu đặt xe |
| 17 | PUT | api/ride/cancelRide/{rideId} | Hủy chuyến đi |
| 18 | PUT | api/ride/completeRide/{rideId} | Hoàn thành chuyến đi |
| 19 | DELETE | api/ride/deleteRide/{rideId} | Xóa chuyến đi |
| 20 | GET | api/ride/history/getRideBooked  /{userId} | Lấy những chuyến đi đã đặt |
| 21 | GET | api/ride/history/getRideCreated  /{userId} | Lấy những chuyến đi đã tạo |
| 22 | POST | api/comment/createComment | Tạo đánh giá |
| 23 | GET | api/ comment /getComments  /{userId} | Trả về danh sách đánh giá của một người dùng |
| 24 | DELETE | api/ comment /deleteComment  /{commentId} | Xóa đánh giá |
| 25 | POST | api/conversation/createConversation |  |
| 26 | GET | api/conversation/getConversation  /{userId} |  |