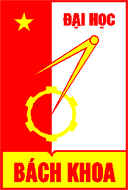


TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

**VIỆN ĐIỆN TỬ - VIỄN THÔNG**

✯✯✯✯✯✯✯✯✯



**ĐỒ ÁN**

**TỐT NGHIỆP ĐẠI HỌC**

**Đề tài:**

**Thiết kế hệ thống bắt xe trực tuyến**

|  |  |
| --- | --- |
| **Sinh viên thực hiện:**  **Số hiệu sinh viên :** | **Vũ Đức Tùng**  **20134459** |
|  |  |
| **Giảng viên hướng dẫn:** | **TS. Phạm Doãn Tĩnh** |

Hà Nội, 1/2018

**Đánh giá quyển đồ án tốt nghiệp**

**(Dùng cho giảng viên hướng dẫn)**

Giảng viên đánh giá: **TS.Phạm Doãn Tĩnh**

Họ và tên Sinh viên: **Vũ Đức Tùng** MSSV: **20134459**

Tên đồ án: ***Thiết kế hệ thống bắt xe trực tuyến***

***Chọn các mức điểm phù hợp cho sinh viên trình bày theo các tiêu chí dưới đây:***

***Rất kém (1); Kém (2); Đạt (3); Giỏi (4); Xuất sắc (5)***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Có sự kết hợp giữa lý thuyết và thực hành (20)** | | | | | | | |
| 1 | Nêu rõ tính cấp thiết và quan trọng của đề tài, các vấn đề và các giả thuyết (bao gồm mục đích và tính phù hợp) cũng như phạm vi ứng dụng của đồ án | | 1 | 2 | 3 | 4 | 5 |
| 2 | Cập nhật kết quả nghiên cứu gần đây nhất (trong nước/quốc tế) | | 1 | 2 | 3 | 4 | 5 |
| 3 | Nêu rõ và chi tiết phương pháp nghiên cứu/giải quyết vấn đề | | 1 | 2 | 3 | 4 | 5 |
| 4 | Có kết quả mô phỏng/thưc nghiệm và trình bày rõ ràng kết quả đạt được | | 1 | 2 | 3 | 4 | 5 |
| **Có khả năng phân tích và đánh giá kết quả (15)** | | | | | | | |
| 5 | Kế hoạch làm việc rõ ràng bao gồm mục tiêu và phương pháp thực hiện dựa trên kết quả nghiên cứu lý thuyết một cách có hệ thống | | 1 | 2 | 3 | 4 | 5 |
| 6 | Kết quả được trình bày một cách logic và dễ hiểu, tất cả kết quả đều được phân tích và đánh giá thỏa đáng. | | 1 | 2 | 3 | 4 | 5 |
| 7 | Trong phần kết luận, tác giả chỉ rõ sự khác biệt (nếu có) giữa kết quả đạt được và mục tiêu ban đầu đề ra đồng thời cung cấp lập luận để đề xuất hướng giải quyết có thể thực hiện trong tương lai. | | 1 | 2 | 3 | 4 | 5 |
| **Kỹ năng viết (10)** | | | | | | | |
| 8 | Đồ án trình bày đúng mẫu quy định với cấu trúc các chương logic và đẹp mắt (bảng biểu, hình ảnh rõ ràng, có tiêu đề, được đánh số thứ tự và được giải thích hay đề cập đến trong đồ án, có căn lề, dấu cách sau dấu chấm, dấu phẩy v.v), có mở đầu chương và kết luận chương, có liệt kê tài liệu tham khảo và có trích dẫn đúng quy định | | 1 | 2 | 3 | 4 | 5 |
| 9 | Kỹ năng viết xuất sắc (cấu trúc câu chuẩn, văn phong khoa học, lập luận logic và có cơ sở, từ vựng sử dụng phù hợp v.v.) | | 1 | 2 | 3 | 4 | 5 |
| **Thành tựu nghiên cứu khoa học (5) (chọn 1 trong 3 trường hợp)** | | | | | | | |
| 10a | | Có bài báo khoa học được đăng hoặc chấp nhận đăng/đạt giải SVNC khoa học giải 3 cấp Viện trở lên/các giải thưởng khoa học (quốc tế/trong nước) từ giải 3 trở lên/ Có đăng ký bằng phát minh sáng chế | 5 | | | | |
| 10b | | Được báo cáo tại hội đồng cấp Viện trong hội nghị sinh viên nghiên cứu khoa học nhưng không đạt giải từ giải 3 trở lên/Đạt giải khuyến khích trong các kỳ thi quốc gia và quốc tế khác về chuyên ngành như TI contest. | 2 | | | | |
| 10c | | Không có thành tích về nghiên cứu khoa học | 0 | | | | |
| **Điểm tổng** | | | **/50** | | | | |
| **Điểm tổng quy đổi về thang 10** | | |  | | | | |

***\* Nhận xét thêm của Thầy/Cô (****giảng viên hướng dẫn nhận xét về thái độ và tinh thần làm việc của sinh viên****)***

...................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

Ngày: / /2018

Người nhận xét

(Ký và ghi rõ họ tên)

**Đánh giá quyển đồ án tốt nghiệp**

**(Dùng cho cán bộ phản biện)**

Giảng viên đánh giá:................................................................................................

Họ và tên Sinh viên: **Vũ Đức Tùng** MSSV: **20134459**

Tên đồ án: ***Thiết kế hệ thống bắt xe trực tuyến***

***Chọn các mức điểm phù hợp cho sinh viên trình bày theo các tiêu chí dưới đây:***

***Rất kém (1); Kém (2); Đạt (3); Giỏi (4); Xuất sắc (5)***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Có sự kết hợp giữa lý thuyết và thực hành (20)** | | | | | | | |
| 1 | Nêu rõ tính cấp thiết và quan trọng của đề tài, các vấn đề và các giả thuyết (bao gồm mục đích và tính phù hợp) cũng như phạm vi ứng dụng của đồ án | | 1 | 2 | 3 | 4 | 5 |
| 2 | Cập nhật kết quả nghiên cứu gần đây nhất (trong nước/quốc tế) | | 1 | 2 | 3 | 4 | 5 |
| 3 | Nêu rõ và chi tiết phương pháp nghiên cứu/giải quyết vấn đề | | 1 | 2 | 3 | 4 | 5 |
| 4 | Có kết quả mô phỏng/thưc nghiệm và trình bày rõ ràng kết quả đạt được | | 1 | 2 | 3 | 4 | 5 |
| **Có khả năng phân tích và đánh giá kết quả (15)** | | | | | | | |
| 5 | Kế hoạch làm việc rõ ràng bao gồm mục tiêu và phương pháp thực hiện dựa trên kết quả nghiên cứu lý thuyết một cách có hệ thống | | 1 | 2 | 3 | 4 | 5 |
| 6 | Kết quả được trình bày một cách logic và dễ hiểu, tất cả kết quả đều được phân tích và đánh giá thỏa đáng. | | 1 | 2 | 3 | 4 | 5 |
| 7 | Trong phần kết luận, tác giả chỉ rõ sự khác biệt (nếu có) giữa kết quả đạt được và mục tiêu ban đầu đề ra đồng thời cung cấp lập luận để đề xuất hướng giải quyết có thể thực hiện trong tương lai. | | 1 | 2 | 3 | 4 | 5 |
| **Kỹ năng viết (10)** | | | | | | | |
| 8 | Đồ án trình bày đúng mẫu quy định với cấu trúc các chương logic và đẹp mắt (bảng biểu, hình ảnh rõ ràng, có tiêu đề, được đánh số thứ tự và được giải thích hay đề cập đến trong đồ án, có căn lề, dấu cách sau dấu chấm, dấu phẩy v.v), có mở đầu chương và kết luận chương, có liệt kê tài liệu tham khảo và có trích dẫn đúng quy định | | 1 | 2 | 3 | 4 | 5 |
| 9 | Kỹ năng viết xuất sắc (cấu trúc câu chuẩn, văn phong khoa học, lập luận logic và có cơ sở, từ vựng sử dụng phù hợp v.v.) | | 1 | 2 | 3 | 4 | 5 |
| **Thành tựu nghiên cứu khoa học (5) (chọn 1 trong 3 trường hợp)** | | | | | | | |
| 10a | | Có bài báo khoa học được đăng hoặc chấp nhận đăng/đạt giải SVNC khoa học giải 3 cấp Viện trở lên/các giải thưởng khoa học (quốc tế/trong nước) từ giải 3 trở lên/ Có đăng ký bằng phát minh sáng chế | 5 | | | | |
| 10b | | Được báo cáo tại hội đồng cấp Viện trong hội nghị sinh viên nghiên cứu khoa học nhưng không đạt giải từ giải 3 trở lên/Đạt giải khuyến khích trong các kỳ thi quốc gia và quốc tế khác về chuyên ngành như TI contest. | 2 | | | | |
| 10c | | Không có thành tích về nghiên cứu khoa học | 0 | | | | |
| **Điểm tổng** | | | **/50** | | | | |
| **Điểm tổng quy đổi về thang 10** | | |  | | | | |

***\* Nhận xét thêm của Thầy/Cô***

................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................ Ngày: / /2018

Người nhận xét

(Ký và ghi rõ họ tên)

# Lời nói đầu

Ngày nay, với sự phát triển của công nghệ thông tin và điện tử truyền thông, các thiết bị điện thoại thông minh ngày càng phổ biến và thiết yếu cho mỗi cá nhân. Người dùng mong muốn các chức năng, dịch vụ nhanh chóng, tin cậy như thanh toán trực tuyến, tìm kiếm trực tuyến, giải trí, bản đồ... Dựa trên cơ sở đó, nhiều hệ thống, ứng dụng được phát triển dành cho điện thoại thông minh và thu được nguồn lợi nhuận lớn nhờ việc đầu tư vốn nhỏ và số lượng người dùng lớn.

Hệ thống bắt xe trực tuyến được xây dựng dành cho đối tượng người dùng sở hữu các thiết bị di động thông minh và có nhu cầu đi lại nhanh chóng, tiện lợi. Lợi dụng các ưu điểm của điện thoại thông minh, hệ thống này có tiềm năng phát triển và mở rộng tốt trong tương lai.

Để thực hiện dự án này, em đã nhận được được sự hướng dẫn nhiệt tình và tận tâm từ TS. Phạm Doãn Tĩnh, em xin chân thành cảm ơn thầy. Do kiến thức hạn hẹp và thời gian nghiên cứu chưa có nhiều nên trong quá trình phân tích, thiết kế hệ thống nếu có gì sai sót mong thầy cô đóng góp ý kiến để em có thể làm đồ án tốt nghiệp của mình hoàn chỉnh hơn nữa.

Hà Nội, ngày 14 tháng 1 năm 2018

Sinh viên thực hiện

**Vũ Đức Tùng**

# Tóm tắt

Trong đề tài này, dựa trên những phương thức, công cụ mới và phổ biến tôi đã xây dựng được hệ thống bắt xe trực tuyến khá hoàn chỉnh. Thêm vào đó, dựa trên việc tìm hiểu các công nghệ khả dụng hiện tại, tôi đã áp dụng các công nghệ phù hợp với yêu cầu và đặc trưng của đề tài.

Đồ án sẽ đưa ra bài toán về xây dựng hệ thống bắt xe trực tuyến, bắt đầu từ khâu phân tích và thiết kế đến khâu triển khai hệ thống. Với việc sử dụng UML (sơ đồ use-case, sơ đồ tuần tự, sơ đồ trạng thái, sơ đồ lớp, sơ đồ thực thế liên kết), mô hình thác nước, các kỹ thuật xử lý thời gian thực, mô hình clients-server và các qui tắc về bảo mật, an toàn thông tin thì tôi đã đạt được một số kết quả.

Đầu tiên, tôi khảo sát về tình hình giao thông vận tải hành khách đô thị và các hệ thống bắt xe trực tuyến có sẵn để tìm ra các đặc điểm, cách thức vận hành, điểm mạnh yếu của các hệ thống. Tiếp đó, tôi lên ý tưởng, đặt ra các mục tiêu, yêu cầu, giới hạn của đề tài, tìm hiểu các qui trình thiết kế và các công nghệ. Sau đó, tôi tập trung vào việc phân tích thiết kế và triển khai hệ thống.

Trong thời gian ba tháng, tôi đã hoàn thành việc khảo sát và tìm hiểu các hệ thống sẵn có, phân tích thiết kế hệ thống, phát triển thuật toán, xây dựng cơ sở dữ liệu và triển khai hệ thống.

# Abstraction

In this project, based on the new methods and tools, I have built a complete online transportation dispatch system. In addition, based on the current availability of technology, i have applied technologies that are appropriate to the needs and characteristics of the topic.

The project will address the problem of building online transportation dispatch system, starting from analysis and design to system deployment. With the use of UML (use-case diagrams, sequence diagram, state diagrams, class diagrams, entity relationship diagram), waterfall models, real-time processing techniques and the rules of security, I have achieved some results.

First of all, I investigated the situation of urban passenger transport and the online transportation dispatch system available to find out the characteristics, how to operate, weak points and strong points of these system. Then, I came up with the idea, set the goals, requiements, limit of the topic, learn the design process and technology. Then, I focused on analyzing the design and deployment of the system.

After a period of three months, I completed a survey and explored available systems, system design analysis, algorithm development, database development, and system deployment.

# Mục lục

[Lời nói đầu 6](#_Toc503888536)

[Tóm tắt 7](#_Toc503888537)

[Abstraction 8](#_Toc503888538)

[Mục lục 9](#_Toc503888539)

[Danh mục hình ảnh 12](#_Toc503888540)

[danh mục bảng biểu 14](#_Toc503888541)

[Thuật ngữ và viết tắt 15](#_Toc503888542)

[Chương 1 Giới thiệu 16](#_Toc503888543)

[1.1 Đặt vấn đề và lý do chọn đề tài 16](#_Toc503888544)

[1.2 Mục tiêu của dự án 16](#_Toc503888545)

[1.3 Phạm vi thực hiện dự án 16](#_Toc503888546)

[Chương 2 Cơ sở lý thuyết 18](#_Toc503888547)

[2.1 Hệ quản trị cơ sở dữ liệu 18](#_Toc503888548)

[2.1.1 MySQL 18](#_Toc503888549)

[2.1.2 Firebase Database 18](#_Toc503888550)

[2.2 Các công nghệ sử dụng phía máy chủ 19](#_Toc503888551)

[2.2.1 Hệ thống truyền thông điệp RabbitMQ 19](#_Toc503888552)

[2.2.2 NodeJS 22](#_Toc503888553)

[2.2.3 Nền tảng Firebase 22](#_Toc503888554)

[2.2.4 Firebase Cloud Function 22](#_Toc503888555)

[2.2.5 Firebase Authentication 23](#_Toc503888556)

[Chương 3 Phân tích thiết kế hệ thống 25](#_Toc503888557)

[3.1 Khảo sát dịch vụ vận tải 25](#_Toc503888558)

[3.1.1 Hoạt động vận tải 25](#_Toc503888559)

[3.1.2 Thông tin về tài xế 25](#_Toc503888560)

[3.1.3 Thông tin về khách hàng 26](#_Toc503888561)

[3.1.4 Thông tin về giá cước các dịch vụ 26](#_Toc503888562)

[3.2 Phân tích và thiết kế tổng thể hệ thống bắt xe trực tuyến 27](#_Toc503888563)

[3.2.1 Các đối tượng tham gia 27](#_Toc503888564)

[3.2.2 Yêu cầu chức năng của hệ thống 27](#_Toc503888565)

[3.2.3 Yêu cầu phi chức năng của hệ thống 28](#_Toc503888566)

[3.2.4 Sơ đồ Use-cases 28](#_Toc503888567)

[3.2.5 Các thành phần của hệ thống 28](#_Toc503888568)

[3.3 Thiết kế thuật toán tìm tài xế 29](#_Toc503888569)

[3.3.1 Xử lý bản đồ thực 29](#_Toc503888570)

[3.3.2 Mô tả quá trình gửi nhận yêu cầu chuyến đi 31](#_Toc503888571)

[3.3.3 Đánh giá thuật toán 32](#_Toc503888572)

[3.4 Thiết kế cơ sỡ dữ liệu 33](#_Toc503888573)

[3.4.1 Phân tích yêu cầu 33](#_Toc503888574)

[3.4.2 Conceptual design 35](#_Toc503888575)

[3.4.3 Logical design 37](#_Toc503888576)

[3.5 Phân tích và thiết kế Ứng dụng bắt xe của khách hàng 38](#_Toc503888577)

[3.5.1 Chức năng 38](#_Toc503888578)

[3.5.2 Các đối tượng tham gia 38](#_Toc503888579)

[3.5.3 Thiết kế chức năng đăng nhập 39](#_Toc503888580)

[3.5.4 Thiết kế chức năng tạo yêu cầu vận chuyển 41](#_Toc503888581)

[3.5.5 Thiết kế chức năng thanh toán 46](#_Toc503888582)

[3.5.6 Thiết kế chức năng xem lịch sử chuyến đi 49](#_Toc503888583)

[3.6 Phân tích và thiết kế ứng dụng nhận chuyến xe của tài xế 50](#_Toc503888584)

[3.6.1 Chức năng 50](#_Toc503888585)

[3.6.2 Các đối tượng tham gia 50](#_Toc503888586)

[3.6.3 Thiết kế chức năng đăng nhập 51](#_Toc503888587)

[3.6.4 Thiết kế chức năng nhận yêu cầu vận chuyển 53](#_Toc503888588)

[3.6.5 Thiết kế chức năng thanh toán 57](#_Toc503888589)

[3.6.6 Thiết kế chức năng xem lịch sử và xem thu nhập 59](#_Toc503888590)

[Chương 4 Triển khai hệ thống 61](#_Toc503888591)

[4.1 Sơ đồ triển khai của hệ thống 61](#_Toc503888592)

[4.2 Ứng dụng bắt xe của khách hàng 62](#_Toc503888593)

[4.3 Ứng dụng nhận chuyến xe của tài xế 66](#_Toc503888594)

[Chương 5 Kết luận và hướng phát triển 70](#_Toc503888595)

[5.1 Kết quả đạt được 70](#_Toc503888596)

[5.2 Phương hướng phát triển 70](#_Toc503888597)

[Tài liệu tham khảo 71](#_Toc503888598)

# Danh mục hình ảnh

[Hình ‎2‑1 Firebase realtime database 19](#_Toc503881388)

[Hình ‎2‑2 Workflow của RabbitMQ 21](#_Toc503881389)

[Hình ‎2‑3 Mô hình competing consumers 21](#_Toc503881390)

[Hình ‎2‑4 Authentication providers 24](#_Toc503881391)

[Hình ‎3‑1 Sơ đồ Use-cases của hệ thống bắt xe trực tuyến 28](#_Toc503881392)

[Hình ‎3‑3 Phương pháp quản lý bản đồ thực 29](#_Toc503881393)

[Hình ‎3‑4 Mối quan hệ giữa hàng đợi và các cell 30](#_Toc503881394)

[Hình ‎3‑5 Qúa trình gửi nhận yêu cầu vận chuyển 31](#_Toc503881395)

[Hình ‎3‑6 Sơ đồ thực thể liên kết 36](#_Toc503881396)

[Hình ‎3‑7 Mô hình cơ sở dữ liệu 37](#_Toc503881397)

[Hình ‎3‑8 Sơ đồ use-cases ứng dụng bắt xe của khách hàng 38](#_Toc503881398)

[Hình ‎3‑9 Sơ đồ phân rã chức năng đăng nhập 39](#_Toc503881399)

[Hình ‎3‑10 Sơ đồ tuần tự của chức năng đăng nhập 40](#_Toc503881400)

[Hình ‎3‑11 Sơ đồ lớp chức năng đăng nhập 41](#_Toc503881401)

[Hình ‎3‑12 Sơ đồ Use-case tạo yêu cầu vận chuyển 41](#_Toc503881402)

[Hình ‎3‑13 Trạng thái của chuyến đi trong quá trình tạo và thực hiện yêu cầu vận chuyển 44](#_Toc503881403)

[Hình ‎3‑14 Mối quan hệ giữa chuyến đi và các đối tượng khác trong chương trình 45](#_Toc503881404)

[Hình ‎3‑15 Sơ đồ use-case chức năng thanh toán 46](#_Toc503881405)

[Hình ‎3‑16 Sơ đồ tuần tự chức năng thêm thẻ thanh toán 47](#_Toc503881406)

[Hình ‎3‑17 Sơ đồ tuần tự quá trình xóa thẻ thanh toán 48](#_Toc503881407)

[Hình ‎3‑18 Sơ đồ lớp chức năng thanh toán chuyến đi 49](#_Toc503881408)

[Hình ‎3‑19 Sơ đồ Use-case chức năng xem lịch sử chuyến đi 49](#_Toc503881409)

[Hình ‎3‑20 Sơ đồ lớp chức năng xem lịch sử chuyến đi 50](#_Toc503881410)

[Hình ‎3‑21 Sơ đồ use-cases ứng dụng nhận chuyến xe của tài xế 51](#_Toc503881411)

[Hình ‎3‑22 Sơ đồ tuần tự quá trình đăng nhập của tài xế 52](#_Toc503881412)

[Hình ‎3‑23 Sơ đồ lớp chức năng đăng nhập 53](#_Toc503881413)

[Hình ‎3‑24 Sơ đồ phân rã chức năng nhận yêu cầu vận chuyển 54](#_Toc503881414)

[Hình ‎3‑25 Trạng thái của tài xế 55](#_Toc503881415)

[Hình ‎3‑26 Sơ đồ lớp chức năng nhận yêu cầu vận chuyển 56](#_Toc503881416)

[Hình ‎3‑27 Sơ đồ phân rã chức năng thanh toán 57](#_Toc503881417)

[Hình ‎3‑28 Trạng thái quá trình thanh toán sau chuyến đi 58](#_Toc503881418)

[Hình ‎3‑29 Sơ đồ lớp chức năng thanh toán sau chuyến đi 59](#_Toc503881419)

[Hình ‎3‑30 Sơ đồ phân rã chức năng xem lịch sử và thu nhập 59](#_Toc503881420)

[Hình ‎3‑31 Sơ đồ lớp chức năng xem lịch sử và xem thu nhập 60](#_Toc503881421)

[Hình ‎3‑2 Sơ đồ triển khai hệ thống 61](#_Toc503881422)

[Hình ‎4‑1 Màn hình đăng nhập 62](#_Toc503881423)

[Hình ‎4‑2 Màn hình xác thực bằng OTP 62](#_Toc503881424)

[Hình ‎4‑3 Màn hình chính 63](#_Toc503881425)

[Hình ‎4‑4 Menu chức năng chính 63](#_Toc503881426)

[Hình ‎4‑5 Thêm thẻ thanh toán 63](#_Toc503881427)

[Hình ‎4‑6 Lịch sử chuyến đ 63](#_Toc503881428)

[Hình ‎4‑7 Chi tiết chuyến đã đi 64](#_Toc503881429)

[Hình ‎4‑8 Màn hình đăng xuất 64](#_Toc503881430)

[Hình ‎4‑9 Tạo chuyến đi mới 64](#_Toc503881431)

[Hình ‎4‑10 Tài xế chấp nhận chuyến đi 64](#_Toc503881432)

[Hình ‎4‑11 Trong chuyến đi 65](#_Toc503881433)

[Hình ‎4‑12 Đánh giá tài xế 65](#_Toc503881434)

[Hình ‎4‑13 Thanh toán qua thẻ thành công 65](#_Toc503881435)

[Hình ‎4‑14 Màn hình đăng nhập 66](#_Toc503881436)

[Hình ‎4‑15 Giao diện chính 66](#_Toc503881437)

[Hình ‎4‑16 Menu chức năng chính 67](#_Toc503881438)

[Hình ‎4‑17 Thống kê thu nhập 67](#_Toc503881439)

[Hình ‎4‑18 Lịch sử chuyến đi 67](#_Toc503881440)

[Hình ‎4‑19 Chi tiết chuyến đã đi 67](#_Toc503881441)

[Hình ‎4‑20 Sắn sàng nhận chuyến 68](#_Toc503881442)

[Hình ‎4‑21 Có chuyến đi được gửi đến 68](#_Toc503881443)

[Hình ‎4‑22 Tài xế đến đón khách 68](#_Toc503881444)

[Hình ‎4‑23 Màn hình trong chuyến đi 68](#_Toc503881445)

[Hình ‎4‑24 Thanh toán qua thẻ 69](#_Toc503881446)

[Hình ‎4‑25 Thanh toán bằng tiền mặt 69](#_Toc503881447)

# 

# danh mục bảng biểu

[Bảng ‎3‑1 Thời gian truyền mỗi bản tin trung bình trên hàng đợi 32](#_Toc503888519)

[Bảng ‎3‑2 Thời gian tìm kiếm cell phụ thuộc lượng bản ghi 33](#_Toc503888520)

# Thuật ngữ và viết tắt

|  |  |
| --- | --- |
| GPS | Global Positioning System |
| OTP | One Time Password |
| UML | Unified Modeling Language |
| HTTP | Hypertext Transfer Protocol |
| JSON | Javascript Object Notation |
| SQL | Structured Query Language |
| API | Application Programing Interface |
| HTTPs | Hypertext Transfer Protocol Secure |
| ack | Acknowledgment |
| NACK | Negative Acknowledgment |

# Giới thiệu

## Đặt vấn đề và lý do chọn đề tài

Cùng với sự bùng nổ của công nghệ thông tin, yêu cầu về nhân lực chất lượng cao để đáp ứng cho ngành này đang là vấn đề được xã hội quan tâm. Các kỹ năng về phân tích thiết kế hệ thống, triển khai hệ thống cần được sinh viên rèn luyện hoàn chỉnh và nghiêm túc để đáp ứng yêu cầu của doanh nghiệp. Ngoài ra, việc tiếp cận với các công nghệ mới cũng là một yếu tố cần thiết do thị trường công nghệ thông tin phát triển và thay đổi từng ngày.

Ngoài lý do triên thì hiện nay sự phổ biến của điện thoại thông minh và nhu cầu đi lại nhanh chóng, an toàn của mọi người tại các thành phố lớn cao, một số công ty đã phát triển hệ thống bắt xe trực tuyến như Uber, Grab. Số lượng người dùng và tài xế lái xe tham gia vào các hệ thống này rất lớn. Tuy nhiên, với giá thành cho chuyến đi khá đắt đỏ, không ổn định và việc khấu trừ tiền của tài xế trong chuyến đi khá lớn, cụ thể hãng Uber trừ 25% tiền từ chuyến đi và Grab trừ 23,6% từ chuyến đi của tài xế khiến cho các hệ thống này đang dần mất niềm tin từ cả phía khách hàng và tài xế lái xe.

Với tình hình như trên, việc phát triển một hệ thống bắt xe trực tuyến giá rẻ cho người dùng và miễn phí cho tài xế lái xe là cần thiết và có khả năng phát triển, mở rộng trong tương lai, đồng thời cũng là cơ hội tốt để hoàn thiện các kỹ năng nghiệp vụ để đáp ứng các yêu cầu của doanh nghiệp.

## Mục tiêu của dự án

Mục tiêu chính của dự án xây dựng, thiết kế hệ thống bắt xe trực tuyến cụ thể gồm các phần sau:

* Tìm hiểu về các công nghệ phát triển phần mềm mới
* Khảo sát, thu thập dữ liệu
* Phân tích thiết kế hệ thống
* Triển khai hệ thống

## Phạm vi thực hiện dự án

Với đề tài này, phạm vi thực hiện được giới hạn trong các vấn đề sau:

* Hệ thống phục vụ người dùng ở các thành phố lớn của Việt Nam như Hà Nội, Hồ Chí Minh
* Hệ thống phục vụ người dùng và tài xế sử dụng thiết bị di động là điện thoại chạy hệ điều hành Android

Kết luận:

Trên đây là những giới thiệu của tôi về hệ thống bắt xe trực tuyến. Chương một đã trình bày được lý do lựa chọn đề tài, mục tiêu thực hiện và phạm vi thực hiện của đề tài.

# Cơ sở lý thuyết

Chương này sẽ giới thiệu tổng quan về các công nghệ được sử dụng và các ứng dụng của chúng trong dự án thiết kế hệ thống bắt xe trực tuyến.

## 2.1 Hệ quản trị cơ sở dữ liệu

### 2.1.1 MySQL

MySQL là hệ quản trị cơ sở dữ liệu mã nguồn mở phổ biến trên thế giới, là một ví dụ cơ bản về hệ quản trị cơ sở dữ liệu quan hệ sử dụng ngôn ngữ truy vấn có cấu trúc (SQL). Ưu điểm của hệ quản trị cơ sở dữ liệu này là tốc độ truy vấn nhanh, ổn định, dễ sử dụng, tính bảo mật cao, có tính khả chuyển, nhẹ, dễ dàng cài đặt, hoạt động trên nhiều hệ điều hành, được hỗ trợ bởi nhiều ngôn ngữ lập trình và cung cấp một hệ thống lớn các hàm tiện ích rất mạnh.

Hệ quản trị cơ sở dữ liệu MySQL sẽ được sử dụng để lưu các dữ liệu quan trọng và cần lưu trữ lâu dài, sử dụng lại nhiều lần.

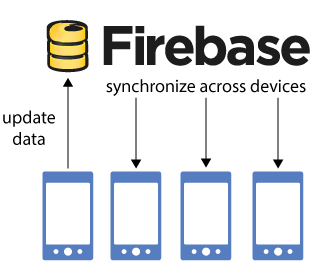
### 2.1.2 Firebase Database

NoSQL là một lớp các hệ cơ sở dữ liệu không sử dụng mô hình quan hệ. Đặc điểm của các hệ cơ sở dữ liệu này là lưu trữ dữ liệu dưới dạng key-value, sử dụng số lượng lớn các node để lưu trữ thông tin, chấp nhận dữ liệu bị trùng lặp do một số node sẽ lưu cùng thông tin giống nhau, không có ràng buộc nào cho việc nhất quán dữ liệu, có hiệu suất và tính sẵn sàng cao

Hệ quản trị cơ sở dữ liệu Firebase là một hệ quản trị dữ liệu NoSQL được lưu trên đám mây. Dữ liệu trong hệ cơ sỡ dữ liệu này được lưu dưới dạng JSON và đồng bộ theo thời gian thực đến mỗi thiết bị kết nối đến

Hệ quản trị cơ sở dữ liệu Firebase có một số năng lực sử dụng chính như sau:

* Thời gian thực: thay vì gửi yêu cầu HTTP, Firebase Database sử dụng đồng bộ hóa dữ liệu, mỗi khi dữ liệu có sự thay đổi, tất cả các thiết bị đang kết nối đến cơ sở dữ liệu sẽ được cập nhật trong thời gian milliseconds
* Chế độ offline: ứng dụng Firebase vẫn hoạt động khi thiết bị bị ngắt kết nối mạng, dữ liệu sẽ được lưu trữ trên thiết bị và sẽ được đồng bộ với cơ sở dữ liệu trên đám mây ngay khi có kết nối mạng trở lại



Hình ‑ Firebase realtime database

* Có thể truy cập trực tiếp từ các thiết bị di động hoặc trình duyệt web mà không cần ứng dụng máy chủ. Tính bảo mật và tin cậy được đảm bảo trong quá trình đọc, ghi và truyền dữ liệu

Với các đặc tính trên, hệ quản trị cơ sở dữ liệu Firebase sẽ được sử dụng để lưu trữ các dữ liệu thay đổi trong thời gian thực, không cần lưu trữ lâu dài.

## 2.2 Các công nghệ sử dụng phía máy chủ

### 2.2.1 Hệ thống truyền thông điệp RabbitMQ

Trong một hệ thống xử lý nghiệp vụ, các ứng dụng có thể được phát triển trên các nền tảng khác nhau, được viết bằng các ngôn ngữ khác nhau, được triển khai trên các thiết bị phần cứng khác nhau thậm chí chạy trên các khoảng thời gian khác nhau, vậy nên, việc giao tiếp giữa các ứng dụng cần linh hoạt và lỏng lẻo, các giao thức giữa các ứng dụng phải đồng nhất. Các hệ thống truyền thông điệp ra đời để đáp ứng các yêu cầu trên.

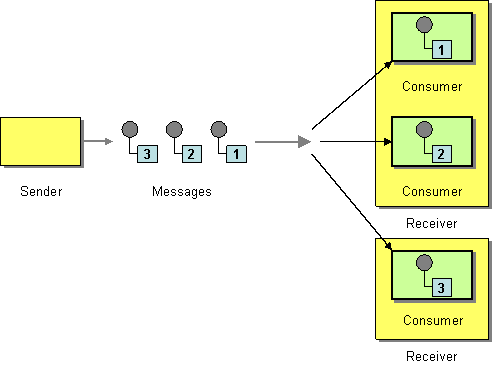
Các khái niệm trong một hệ thống truyền thông điệp

* + Kênh thông điệp (channels): các ứng dụng truyền thông điệp qua các kênh thông điệp, một đường ống ảo kết nối giữa bên gửi và bên nhận. Một hệ thống truyền thông điệp không chứa các kênh mà việc tạo ra kênh truyền dựa trên yêu cầu giao tiếp giữa các ứng dụng.
  + Thông điệp (messages): là một gói thông tin duy nhất có thể truyền trên kênh thông điệp. Một ứng dụng muốn truyền dữ liệu, nó phải chia dữ liệu ra thành các gói, sau đó đóng mỗi gói lại thành một thông điệp và gửi thông điệp lên kênh truyền. Tương tự một ứng dụng muốn nhận dữ liệu phải trích xuất dữ liệu từ các thông điệp và xử lý chúng. Hệ thống truyền thông điệp sẽ gửi nhiều lần các thông điệp đến khi thông được gửi thành công.
  + Truyền dẫn nhiều bước (multi-step delivery): trong trường hợp đơn giản nhất, hệ thống truyền thông điệp sẽ gửi thông điệp trực tiếp từ máy tính của người gửi đến máy tính của người nhận. Tuy nhiên các hoạt động quan trọng thường diễn ra trên quá trình truyền thông điệp sau khi gửi và trước khi nhận.
  + Điều hướng (routing): trong một hệ thống nghiệp vụ lớn gồm nhiều ứng dụng và các kênh truyền kết nối chúng, một thông điệp có thể đi qua nhiều kênh trước khi đến được đích cuối cùng. Quá trình điều hướng cho thông điệp có thể phức tạp mà bên gửi không biết được rằng thông điệp có đi đến đích cuối cùng theo ý muốn hay không. Thay vào đó, bên gửi sẽ gửi thông điệp qua các bộ định tuyến thông điệp, một thành phần có chức năng chuyển hướng giao thức kênh truyền và chuyển thông điệp đến bên nhận cuối cùng hoặc bộ định tuyến thông điệp tiếp theo.
  + Kênh truyền điểm đến điểm (point-to-point channel): việc gửi thông điệp trên một kênh chuyền điểm đến điểm đảm bảo rằng chỉ có một bên nhận nhận được một thông điệp nhất định. Nếu kênh truyền có rất nhiều bên nhận khác nhau thì chỉ một trong số chúng có thể xử lý một thông điệp nhất định. Nếu có rất nhiều bên nhận cố gắng xử lý chung một thông điệp thì kênh truyền sẽ đảm bảo rằng chỉ một bên nhận thành công.
  + Producer: bên gửi thông điệp, là bên tạo ra thông điệp và gửi thông điệp đến hệ thống truyền thông điệp.
  + Consumer: bên nhận thông điệp, là nơi cuối cùng thông điệp đi đến và được xử lý.
  + Message Broker là một chương trình trung gian có tác dụng chuyển một thông điệp từ giao thức chuẩn của bên gửi sang giao thức chuẩn của bên nhận. Message Broker là một thành phần trong viễn thông hay mạng máy tính nơi mà ứng dụng phần mềm giao tiếp với nhau qua các thông điệp được chuẩn hóa.



Hình ‑ Workflow của RabbitMQ

RabbitMQ là một Message Broker sử dụng giao thức AMQP, được lập trình bằng ngôn ngữ Erlang. RabbitMQ cung cấp phương tiện để giao tiếp giữa các thành phần trong một hệ thống lớn, bằng cách nhận thông điệp từ bên gửi sau đó lưu trữ an toàn và gửi đến bên nhận.



Hình ‑ Mô hình competing consumers

Mô hình Competing Consumer tạo ra nhiều consumers trên một kênh truyền đơn để các consumers có thể xử lý nhiều thông điệp một cách đồng thời. Khi một kênh truyền thông điệp chuyển thông điệp đi, bất kỳ consumer nào cũng có thể nhận được thông điệp. Việc triển khai hệ thống truyền thông điệp sẽ quyết định xem consumer nào thực sự nhận được thông điệp. Mỗi Competing Consumer chạy trên một luồng riêng hoặc một thiết bị riêng nên chúng có thể xử lý nhiều thông điệp một cách đồng thời. Trong khi một Consumer đang xử lý thông điệp, kênh truyền có thể gửi một thông điệp khác cho một Consumer khác.

### 2.2.2 NodeJS

NodeJS là một nền tảng phía máy chủ được xây dựng trên Javascript Engine (V8 Engine). NodeJS được dùng để xây dựng các ứng dụng hướng sự kiện chạy bên phía máy chủ, được xây dựng theo mô hình non-blocking I/O để tạo ra các ứng dụng nhẹ và hiệu quả phục vụ cho các ứng dụng thời gian thực chạy trên các thiết bị phân tán. Đặc điểm của NodeJS:

* Không đồng bộ và hướng sự kiện
* Tốc độ chạy nhanh
* Các tiến trình đơn giản nhưng hiệu năng cao
* Không lưu trữ dữ liệu buffer
* Có hệ thống thư viện hỗ trợ lập trình phong phú

### 2.2.3 Nền tảng Firebase

Firebase là một nền tảng cho phép xây dựng các ứng dụng di động một cách nhanh chóng, nó cung cấp các dịch vụ có sẵn như: Realtime Database (cơ sở dữ liệu thời gian thực), Firebase Authentication (xác thực người dùng), Cloud Function (chức năng), Cloud Store (lưu trữ file), Google Analytic (phân tích dữ liệu), Admob (quảng cáo)... Firebase cung cấp một hệ thống khá toàn diện các chức năng để phục vụ phát triển ứng dụng di động, cung cấp cho nhà phát triển các API để phát triển ứng dụng, nhà phát triển sẽ chỉ cần tập trung phát triển ứng dụng về logic, trải nghiệm người dùng, nội dung... Còn các việc xử lý phức tạp bên phía máy chủ (duy trì máy chủ, duy trì cơ sở dữ liệu, xác thực người dùng...) sẽ được xử lý một cách tự động

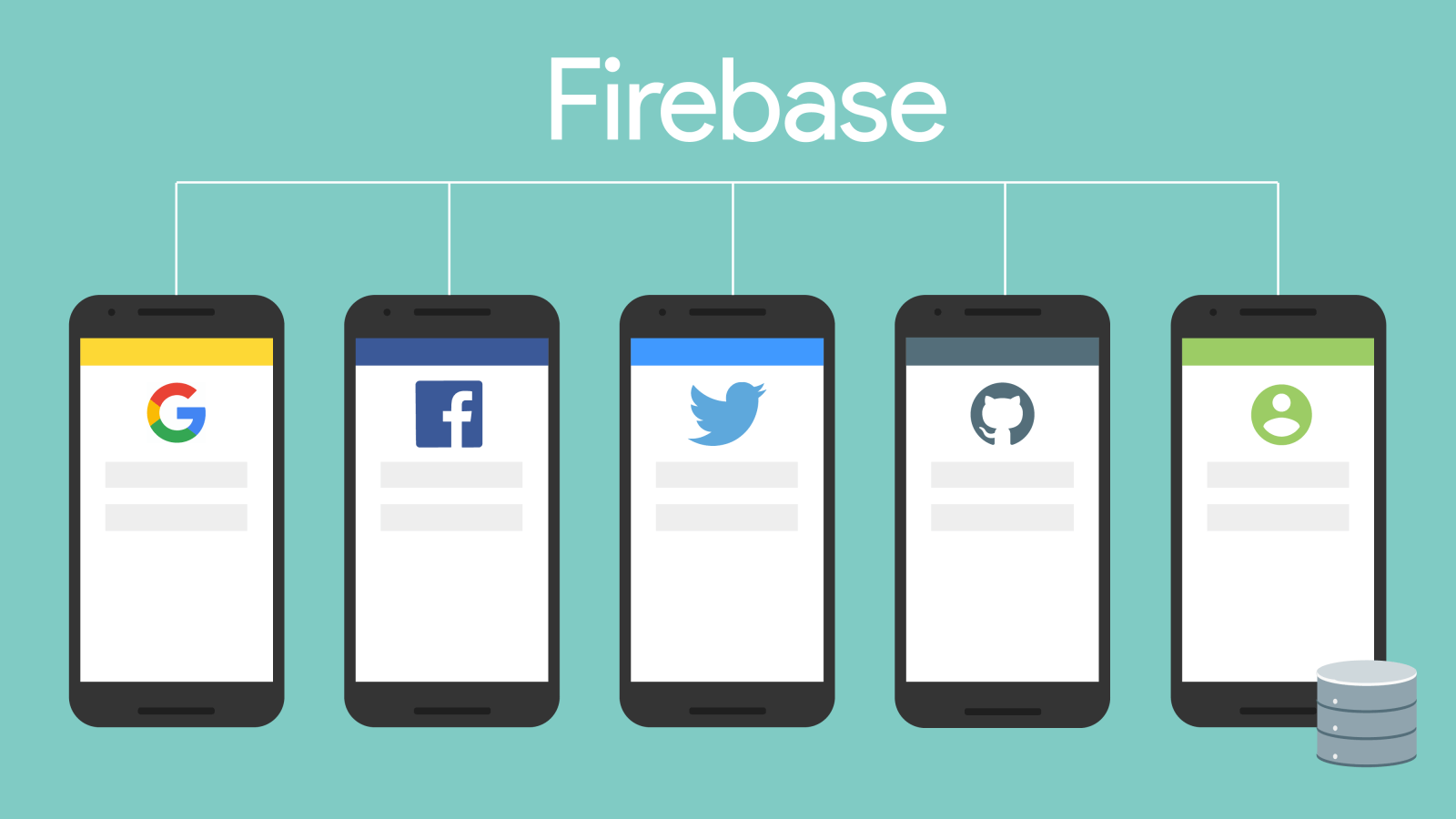
### 2.2.4 Firebase Cloud Function

Cloud Function cho phép chạy mã nguồn phía máy chủ một cách tự động để phản hồi lại các sự kiện được kích hoạt bởi các thành phần khác của Firebase hoặc yêu cầu HTTPS. Mã nguồn ứng dụng phía máy chủ sẽ được lưu trên đám mây của Google và được chạy trên một môi trường được quản lý, nhà phát triển sẽ không phải quan tâm đến việc duy trì và cân bằng máy chủ.

* Các khả năng chính của Cloud Function:
* Hợp nhất với nền tảng Firebase: các chức năng được nhà phát triển xây dựng có thể phản hồi lại các sự kiện sinh ra bởi các thành phần khác của nền tảng Firebase như Cloud Store, Realtime Database, Firebase Authentication, Google Analytics, HTTP Triggers...
* Nhà phát triển không cần duy trì máy chủ: nhà phát triển sẽ triển khai mã nguồn Javascript lên máy chủ sử dụng của sổ dòng lệnh, sau đó Firebase sẽ tự động cân bằng tài nguyên phù hợp với số lượng người dùng. Nhà phát triển không cần quan tâm về cài đặt máy chủ, nâng cấp, bảo trì, giất phép...
* Bảo mật: Cloud Function hoàn toàn tách biệt với phía máy trạm, mã nguồn sẽ được lưu trữ an toàn, bảo mật và đảm bảo thực hiện đúng các chức năng mà nhà phát triển đã thiết kế.
* Vòng đời của một chức năng (function):
* Nhà phát triển viết mã nguồn cho một function mới, lựa chọn nguồn phát sinh sự kiện (event provider) như HTTPS, Realtime Database..., và định nghĩa điều kiện để function được thực hiện
* Nhà phát triển triển khai function và Firebase sẽ kết nối chức năng đó với nguồn phát sinh sự kiện được lựa chọn
* Khi nguồn phát sinh sự kiện tạo ra một sự kiện khớp với điều kiện của function, mã nguồn sẽ được thực hiện
* Nếu function đang bận xử lý nhiều sự kiện, Google sẽ tạo ra thêm các instance để xử lý công việc nhanh hơn. Nếu function rảnh rỗi thì các instance sẽ được xóa bỏ
* Khi nhà phát triển cập nhật function bằng cách triển khai thêm mã nguồn mới, tất cả các instance của phiên bản cũ sẽ bị loại bỏ và thay thế bằng các instance mới
* Khi nhà phát triển xóa bỏ function, tất cả các instance sẽ được xóa bỏ, và kết nối giữa function và nguồn phát sinh sự kiện cũng bị xóa bỏ theo

### 2.2.5 Firebase Authentication

Firebase Authentication được phát triển nhằm mục đích xây dựng các hệ thống xác thực bảo mật một cách dễ dàng, trong khi cải thiện trải nghiệm người dùng trong quá trình đăng nhập, xác thực... Firebase Authentication cung cấp giải pháp end-to-end toàn diện, hỗ trợ đăng nhập bằng email, số điện thoại, tài khoản Google, Facebook, Twitter, Github...



Hình ‑ Authentication providers

Các đặc điểm của Firebase Authentication:

* Xác thực dựa trên email và mật khẩu, Firebase Authentication SDK cung cấp các phương thức để tạo và quản lý người dùng sử dụng địa chỉ email và mật khẩu để đăng nhập. Firebase Authentication xử lý cả việc gửi email cài đặt lại mật khẩu
* Xác thực người dùng bằng việc sử dụng các hệ thống khác như thông qua tài khoản Google, Facebook, Twitter và Github
* Xác thực người dùng qua số điện thoại bằng cách gửi tin nhắn cho người dùng
* Xác thực nặc danh

Kết luận:

Chương 2 dựa trên việc tìm hiểu các công nghệ mới và sẵn có hiện tại, tôi đã lựa chọn được các công nghệ phù hợp và trình bày ngắn gọn về các công nghệ được sử dụng trong đề tài cũng như các điểm mạnh của chúng.

# Phân tích thiết kế hệ thống

Chương này bao gồm toàn bộ các khảo sát, thu thập dữ liệu, phân tích và thiết kế hệ thống từ tổng thể hệ thống đến chi tiết từng thành phần. Mỗi thành phần sẽ được phân tích và thiết kế chi tiết từ chức năng cho đến sơ đồ lớp.

## 3.1 Khảo sát dịch vụ vận tải

Chương này sẽ tổng hợp những khảo sát trong thực tế về dịch vụ vận tải hành khách trong nội thành Hà Nội, bao gồm thông tin về hoạt động vận tải, thông tin về tài xế lái xe, khách hàng và giá cước dịch vụ của một số công ty vận tải.

### 3.1.1 Hoạt động vận tải

Hoạt động vận tải bằng xe ôm và taxi diễn ra như sau:

* Tài xế sở hữu phương tiện giao thông sẽ đợi ở các nơi công cộng như bến xe khách, điểm dừng xe bus, các ngã tư...
* Khách hàng khi có nhu cầu đi lại sẽ tìm tài xế ở các nơi công cộng kể trên (nếu dùng dịch vụ xe ôm), gọi điện đến tổng đài các hãng taxi để yêu cầu xe hoặc đợi các xe taxi đi qua (nếu dùng dịch vụ taxi)
* Khi khách hàng tìm được tài xế, khách hàng sẽ cho tài xế thông tin về điểm đến và tài xế sẽ cho khách hàng thông tin về giá cả. Nếu cả khách hàng và tài xế đã đồng thuận về giá cả và điểm đến thì chuyến đi sẽ được bắt đầu
* Khi đến điểm đến do khách hàng yêu cầu, khách hàng thanh toán tiền cho tài xế (thường sử dụng tiền mặt) và kết thúc chuyến đi

### 3.1.2 Thông tin về tài xế

Các thông tin về tài xế lái xe bao gồm:

* Họ tên
* Số điện thoại
* Loại dịch vụ (xe máy, taxi)
* Thông tin về phương tiện: loại phương tiện, biển số, mô tả bên ngoài
* Địa điểm làm việc

Đối tượng: những người sở hữu phương tiện giao thông, có thời gian nhàn rỗi và có nhu cầu kiếm thêm thu nhập như sinh viên, nhân viên nghỉ hưu, những người chưa có việc làm ổn định...

### 3.1.3 Thông tin về khách hàng

Các thông tin về khách hàng bao gồm:

* Họ tên
* Số điện thoại
* Loại dịch vụ muốn sử dụng

Đối tượng: là những người có nhu cầu đi lại và cần dịch vụ nhanh, an toàn.

### 3.1.4 Thông tin về giá cước các dịch vụ

Giá cước dịch vụ xe ôm theo khảo sát như sau:

* Xe ôm truyền thống:
  + Khung giờ bình thường:
    - km đầu – km 6: 6.000 vnđ/km
    - km 7 – km 12: 5.000 vnđ/km
    - km 13 trở đi: 4.000 vnđ/km
* Xe ôm Grabike:
  + Khung giờ bình thường:
    - 2 km đầu: 11.000 vnđ/km
    - Sau 2 km đầu: 3.800 vnđ/km
  + Khung giờ từ 0h sáng đến 5h sáng và khung giờ cao điểm cộng thêm 10.000 vnđ cả chuyến
* Xe ôm Uber:
  + Khung giờ bình thường:
    - 3.700 vnđ/km
    - Cước phí tối thiểu 10.000 vnđ/km
    - Cước phí hủy chuyến: 5.000 vnđ
  + Khung giờ cao điểm hoặc trời mưa thì giá sẽ được nhân hệ số

Giá cước các dịch vụ taxi theo khảo sát như sau:

* Taxi Mai Linh 4 chỗ:
  + Giá mở cửa: 10.000 vnđ/0.75km
  + Km tiếp theo: 13.600 vnđ/km
  + Km thứ 31 trở đi: 11.000 vnđ/km
* Taxi Vinasun 4 chỗ:
  + Giá mở cửa: 11.000 vnđ/500m
  + Trong phạm vi 30km: 14.500 vnđ/km
  + Km thứ 31 trở đi: 11.600 vnđ/km
* Taxi Grab 4 chỗ:
  + Khung giờ bình thường:
    - Giá tối thiểu: 20.000 vnđ
    - 2 km đầu: 10.000 vnđ/km
    - Km tiếp theo: 8.500 vnđ/km
  + Khung giờ cao điểm: nhân giá khung giờ bình thường với hệ số
* Taxi Uber phổ thông:
  + Khung giờ bình thường:
    - Giá tối thiểu: 15.000 vnđ/2 km đầu
    - Km tiếp theo: 7.500 vnđ/km
    - Cước phí hủy chuyến: 15.000 vnđ
  + Khung giờ cao điểm: nhân giá khung giờ bình thường với hệ số

## 3.2 Phân tích và thiết kế tổng thể hệ thống bắt xe trực tuyến

### 3.2.1 Các đối tượng tham gia

Các đối tượng tham gia vào hệ thống bắt xe trực tuyến bao gồm:

* Tài xế lái xe
* Khách hàng sử dụng dịch vụ
* Dịch vụ thanh toán để phục vụ khách hàng thanh toán qua thẻ ngân hàng và trả tiền lương cho tài xế lái xe
* Admin quản lý các thông tin về tài xế lái xe và giá cước của dịch vụ

### 3.2.2 Yêu cầu chức năng của hệ thống

Yêu cầu chức năng của hệ thống bắt xe trực tuyến bao gồm:

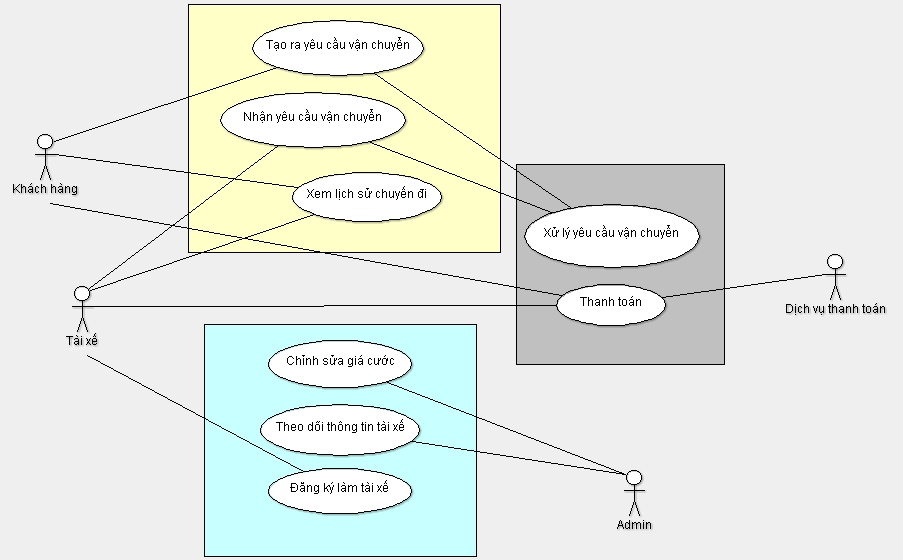
* Cho phép tài xế đăng ký tham gia vào hệ thống
* Nhận yêu cầu vận chuyển từ khách hàng
* Xử lý yêu cầu vận chuyển của khách hàng, tìm tài xế phù hợp với khách hàng
* Gửi yêu cầu vận chuyển cho tài xế
* Nhận tiền thanh toán từ khách hàng và trả lương cho tài xế
* Cho khách hàng và tài xế xem lại lịch sử các chuyến xe đã đi
* Cho phép admin duyệt hồ sơ đăng ký của tài xế và chỉnh sửa giá cước của dịch vụ

### 3.2.3 Yêu cầu phi chức năng của hệ thống

Yêu cầu phi chức năng của hệ thống bắt xe trực tuyến bao gồm:

* Thời gian xử lý yêu cầu vận chuyển nhanh và tin cậy
* Giao diện thân thiện, dễ sử dụng cho cả khách hàng và tài xế
* Xử lý thanh toán phải nhanh và đảm bảo tính bảo mật, tin cậy

### 3.2.4 Sơ đồ Use-cases



Hình ‑ Sơ đồ Use-cases của hệ thống bắt xe trực tuyến

### 3.2.5 Các thành phần của hệ thống

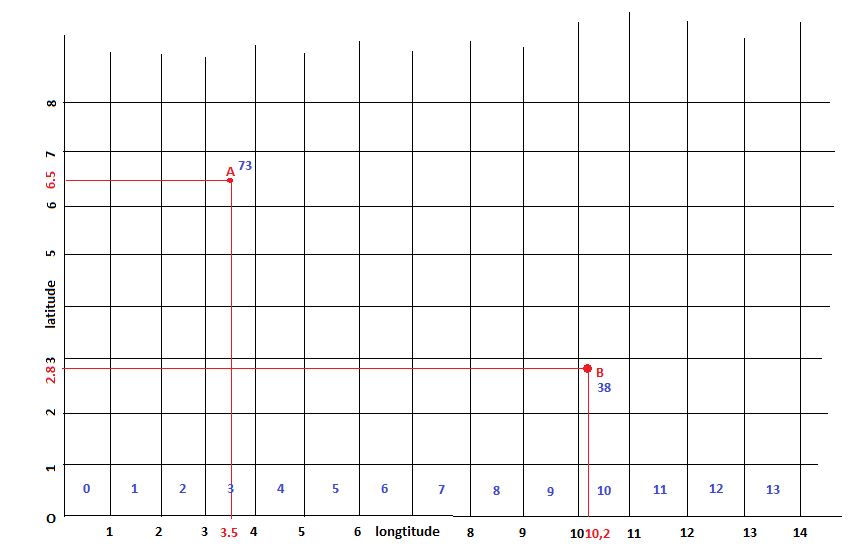
Các thành phần của hệ thống bắt xe trực tuyến bao gồm:

* Ứng dụng bắt xe của khách hàng
* Ứng dụng nhận chuyến xe của tài xế
* Ứng dụng máy chủ nghiệp vụ xử lý yêu cầu vận chuyển và thanh toán
* Ứng dụng máy chủ web dùng để đăng ký thông tin cho tài xế
* Ứng dụng quản lý dành cho quản trị viên

## 3.3 Thiết kế thuật toán tìm tài xế

### 3.3.1 Xử lý bản đồ thực

* + Ý tưởng: chia bản đồ các khu vực thành các ô vuông nhỏ (cell), sau đó đánh số các cell và quản lý mỗi cell, mỗi khách hàng hoặc tài xế chỉ có thể đặt chuyến hoặc nhận chuyến trong cell đó.
  + Ví dụ ở hình 5-2 bên dưới, chia khu vực này thành 112 cell vuông bằng nhau và đánh số mỗi ô từ 0-111. Khi đó điểm A có tọa độ (3.5, 6.5) sẽ ở cell có số 73, điểm B có tọa độ (10.2, 2.8) sẽ ở cell có số 38.

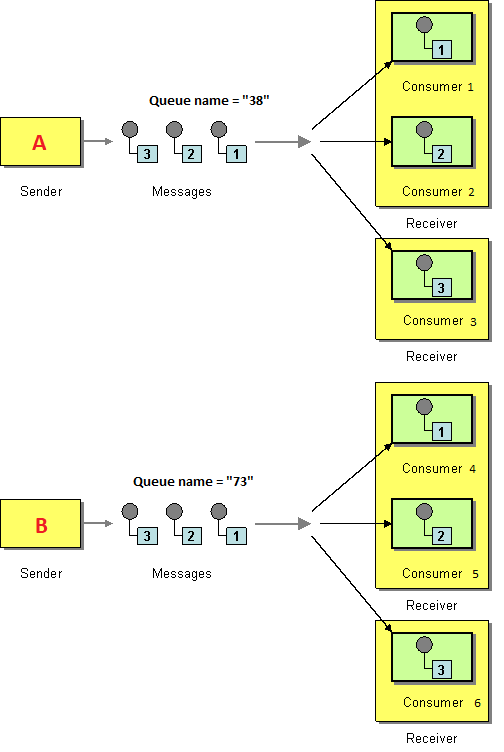


Hình ‑ Phương pháp quản lý bản đồ thực

Giả sử có một khách hàng ở điểm A đặt chuyến, kênh truyền thông điệp sẽ được đặt tên “73”, khách hàng ở điểm B đặt chuyến thì kênh truyền thông điệp sẽ được đặt tên “38”. Như vậy ta có các sơ đồ như hình 5-3 bên dưới.

Nếu có các tài xế ở các cell tương ứng, ví dụ tài xế 1, 2, 3 ở cell số 38, họ sẽ chỉ nhận được các yêu cầu của khách hàng ở cell số 38 trên kênh thông điệp “38”. Tương tự với các tài xế 4, 5, 6 ở cell số 73 cũng sẽ chỉ nhận được các yêu cầu đặt chuyến của khách hàng ở cell số 73 trên kênh thông điệp “73”.

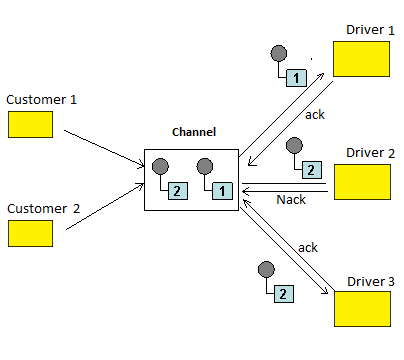
Do các cell là độc lập và duy nhất, nên các kênh thông điệp cũng sẽ là độc lập và duy nhất. Các khách hàng ở cell số 38 sẽ không thể gọi được tài xế ở cell số 73 và ngược lại.



Hình ‑ Mối quan hệ giữa hàng đợi và các cell

Việc tìm kiếm cell hiện tại của khách hàng hoặc tài xế sẽ được xử lý trên máy chủ nghiệp vụ. Mỗi khu vực sẽ có số lượng cell khác nhau do diện tích khác nhau, và các cell là độc lập và duy nhất trên toàn thế giới.

### 3.3.2 Mô tả quá trình gửi nhận yêu cầu chuyến đi



Hình ‑ Qúa trình gửi nhận yêu cầu vận chuyển

Hình 5-4 mô tả quá trình gửi nhận yêu cầu vận chuyển giữa khách hàng và tài xế lái xe. Khi người dùng 1 và người dùng 2 gửi yêu cầu đặt chuyến đến kênh thông điệp, kênh thông điệp sẽ tìm các tài xế lái xe đang rảnh rỗi, ví dụ trên hình 5-4, kênh truyền thông điệp sẽ gửi thông điệp nhận chuyến đến tài xế 1 và tài xế 2. Tài xế 1 trong trường hợp này chấp nhận chuyến đi, ứng dụng nhận chuyến sẽ phản hồi lại bản tin ack (acknowledgment) đến hàng đợi, lúc này hàng đợi sẽ mặc định rằng thông điệp thứ nhất đã được thực hiện. Tài xế 2 trong trường hợp này không chấp nhận chuyến đi, ứng dụng nhận chuyến của tài xế 2 sẽ gửi lại bản tin Nack (negative acknowledgment) đến kênh thông điệp. Kênh thông điệp sẽ tiếp tục tìm tài xế đang rảnh rỗi, trong trường hợp này là tài xế 3, tài xế 3 chấp nhận chuyến đi và ứng dụng nhận chuyến của tài xế 3 sẽ gửi lại bản tin ack xác nhận rằng thông điệp đã được xử lý. Sau khi đã nhận chuyến đi, ứng dụng nhận chuyến của tài xế lái xe sẽ ngừng theo dõi trên kênh thông điệp cho đến thời điểm chuyến đi kết thúc, tài xế sau đó sẽ theo dõi thông điệp trên kênh thông điệp khác và ở một cell khác.

### 3.3.3 Đánh giá thuật toán

Thời gian thực hiện thuật toán là thời gian tìm kiếm cell hiện tại của tài xế và khách hàng cộng với thời gian bản tin được gửi đi từ ứng dụng của khách hàng đến ứng dụng của tài xế, tức là thời gian bản tin được gửi đi trên hàng đợi.

Các kết quả đánh giá được thể hiện trong các bảng bên dưới, cấu hình máy tính chạy đánh giá như sau:

* + Vi xử lý: Intel Core i5-4210U CPU @ 1.7GHz up to 2.4GHz
  + RAM: 8 GB
  + Hệ thống 64-bit, x64-base processor

Bảng ‑ Thời gian truyền mỗi bản tin trung bình trên hàng đợi

|  |  |  |  |
| --- | --- | --- | --- |
| Số lượng hàng đợi | Số lượng người dùng trên một hàng đợi | Tổng số người dùng | Thời gian trung bình (ms) |
| 50 | 10 | 500 | < 1 |
| 100 | 10 | 1.000 | < 1 |
| 200 | 10 | 2.000 | < 1 |
| 500 | 5 | 2.500 | < 1 |
| 300 | 10 | 3.000 | < 1 |
| 200 | 20 | 4.000 | < 1 |
| 1.000 | 5 | 5.000 | < 1 |
| 500 | 10 | 5.000 | < 1 |
| 1.000 | 100 | 100.000 | < 1 |
| 10.000 | 10 | 100.000 | 1.27 |

Bảng ‑ Thời gian tìm kiếm cell phụ thuộc lượng bản ghi

|  |  |
| --- | --- |
| Số lượng bản ghi | Thời gian tìm cell (ms) |
| 2500 | 0 |
| 5000 | 0 |
| 13000 | 15 |
| 20000 | 16 |
| 40000 | 31 |
| 61000 | 47 |
| 140000 | 375 |
| 326000 | 750 |

## 3.4 Thiết kế cơ sỡ dữ liệu

### 3.4.1 Phân tích yêu cầu

Cơ sở dữ liệu của hệ thống bắt xe trực tuyến sẽ lưu thông tin về các chuyến đi, khách hàng, tài xế, quản trị viên, các thông tin về từng khu vực và giá cước dịch vụ. Thông tin về các thực thể lưu trong cơ sở dữ liệu:

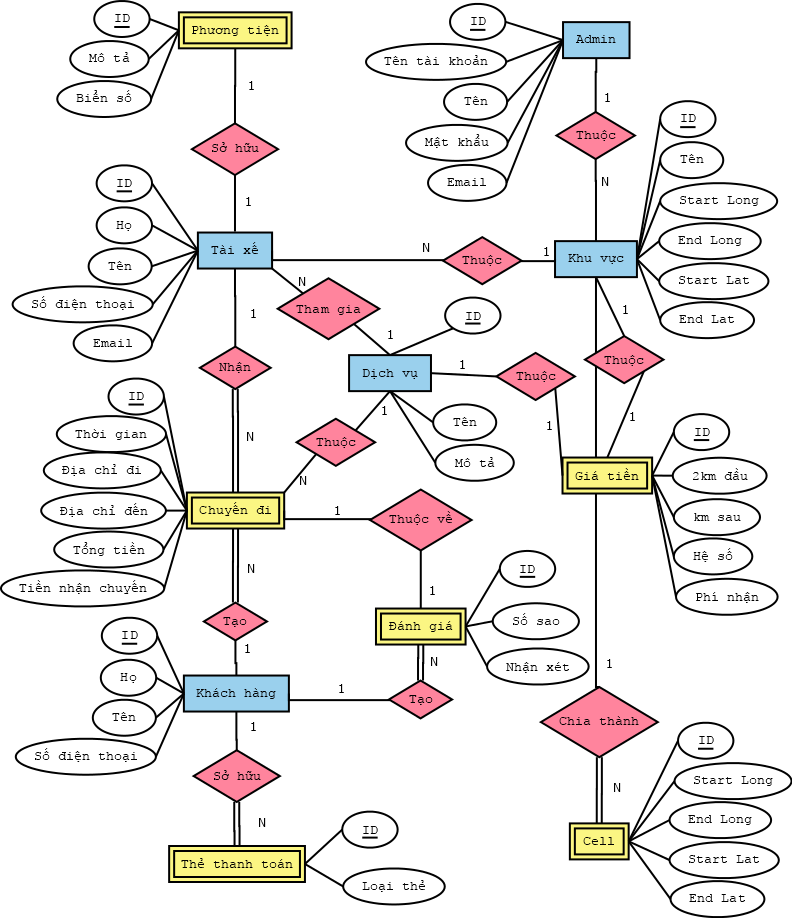
* Một khách hàng có thể có trong cơ sỡ dữ liệu mà không phụ thuộc vào các đối tượng khác, do đó khách hàng là một thực thể mạnh. Thông tin về khách hàng bao gồm: id của khách hàng để phân biệt với các khách hàng khác, họ tên, số điện thoại, email, thẻ thanh toán (debit/credit card)
* Thẻ thanh toán là một thực thể yếu vì nó tồn tại phụ thuộc vào khách hàng. Thông tin về thẻ thanh toán bao gồm mã số đăng ký thẻ trên trang web <https://www.simplify.com> và id để phân biệt với các thẻ thanh toán khác
* Khu vực là một thực thể mạnh vì một khu vực có thể tồn tại mà không phụ thuộc vào admin, khách hàng hay tài xế... Thông tin về khu vực bao gồm id khu vực để phân biệt với khu vực khác, tên khu vực, tọa độ bắt đầu và tọa độ kết thúc, giá tiền của khu vực
* Cell là thực thể yếu vì một cell thuộc về một khu vực nào đó, không thể có thông tin về cell nếu không có thông tin về khu vực, thông tin về cell bao gồm tọa độ bắt đầu, tọa độ kết thúc và id để phân biệt giữa các cell
* Dịch vụ là một thực thể mạnh vì nó có thể tồn tại độc lập trong cơ sở dữ liệu. Dịch vụ có các thuộc tính: id dịch vụ để phân biệt với dịch vụ khác, tên của dịch vụ và mô tả chi tiết về dịch vụ
* Tài xế là một thực thể mạnh vì nó có thể tồn tại độc lập trong cơ sở dữ liệu, các thông tin về tài xế bao gồm: họ tên, công việc hiện tại, ngày tháng năm sinh, số điện thoại, địa chỉ email, thông tin về phương tiện
* Phương tiện giao thông là một thực thể yếu vì một phương tiện giao thông phải thuộc về một tài xế nào đó, sẽ không có thông tin về phương tiện giao thông mà không có thông tin về tài xế
* Chuyến đi là một thực thể yếu vì chỉ có thông tin về chuyến đi khi khách hàng yêu cầu, thông tin về chuyến đi bao gồm: id phân biệt giữa các chuyến đi, thời gian yêu cầu, địa điểm bắt đầu, địa điểm kết thúc, giá tiền khoảng cách, trạng thái (thành công, bị hủy...) và số tiền tài xế nhận chuyến
* Đánh giá là một thực thể yếu vì một đánh giá thuộc về một khách hàng và chỉ một chuyến đi, một đánh giá bao gồm id để phân biệt với đánh giá khác, số sao và nhận xét của khách hàng

Thông tin về các quan hệ:

* Một khu vực có thể có nhiều cell, nhưng mỗi cell chỉ thuộc về một khu vực, do đó quan hệ giữa khu vực và cell là quan hệ 1:N
* Một tài xế chỉ hoạt động ở một khu vực và một khu vực có thể có nhiều tài xế, một chỉ được đăng ký một phương tiện giao thông và mỗi phương tiện giao thông thuộc về một tài xế, do đó quan hệ giữa tài xế và khu vực là N:1, quan hệ giữa tài xế và phương tiện giao thông là 1:1
* Một khách hàng có thể có nhiều thẻ thanh toán nhưng mỗi thẻ thanh toán chỉ thuộc về một khách hàng, do đó quan hệ giữa thẻ thanh toán và khách hàng là quan hệ N:1
* Một đánh giá chỉ thuộc về một khách hàng và một khách hàng có thể đánh giá nhiều lần, do đó quan hệ giữa đánh giá và khách hàng là quan hệ N:1, trong đó một đánh giá chỉ thuộc về một chuyến đi và một chuyến đi có tối đa một đánh giá, do đó quan hệ giữa đánh giá và chuyến đi là quan hệ 1:1
* Một chuyến đi do một khách hàng tạo ra, nhưng một khách hàng có thể yêu cầu nhiều chuyến đi, do đó quan hệ giữa khách hàng và chuyến đi là quan hệ 1:N. Một chuyến đi chỉ do một tài xế nhận và một tài xế có thể nhận nhiều chuyến đi, do đó quan hệ giữa tài xế và chuyến đi là quan hệ 1:N

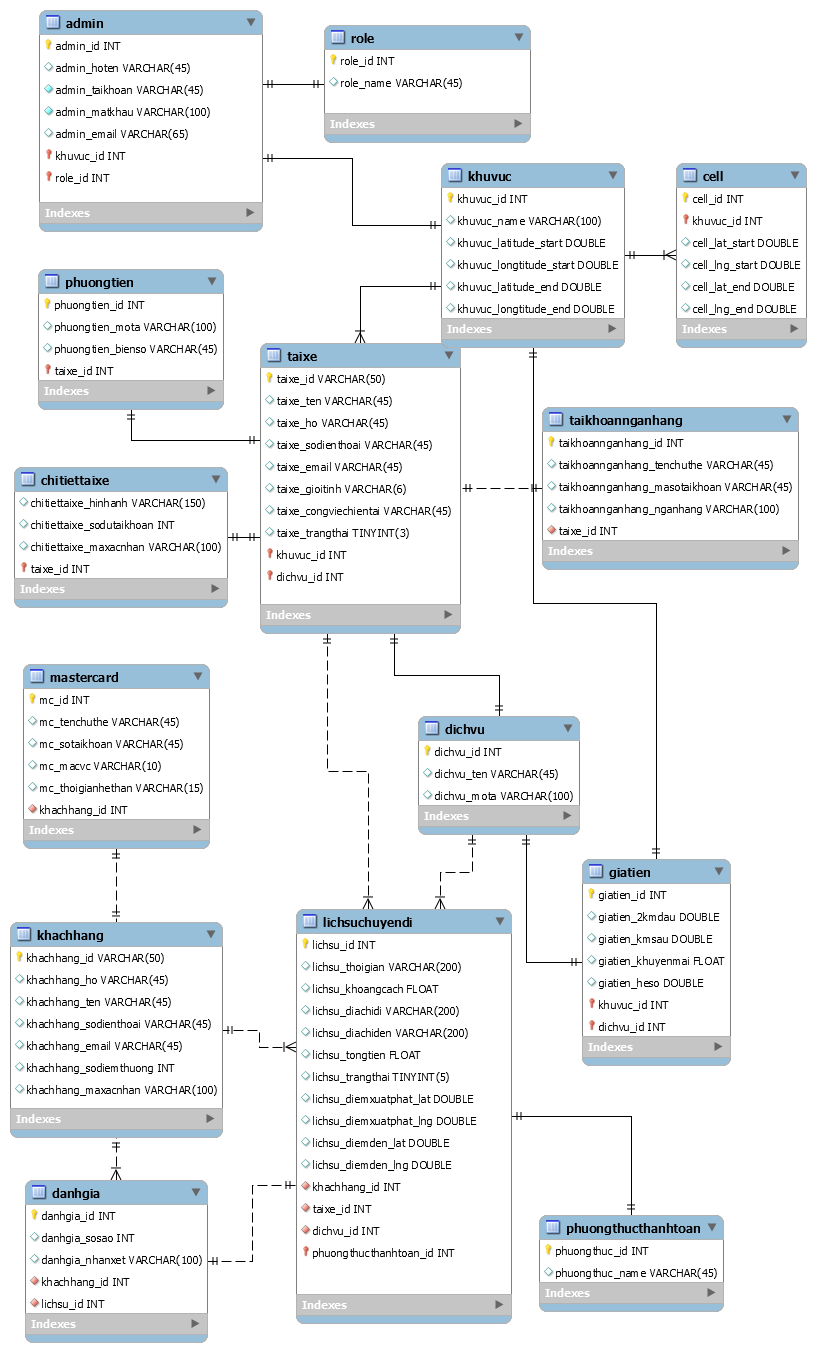
### 3.4.2 Conceptual design

Sơ đồ thực thể liên kết hình 3-6 dựa trên phân tích yêu cầu của cơ sở dữ liệu.



Hình ‑ Sơ đồ thực thể liên kết

### 3.4.3 Logical design



Hình ‑ Mô hình cơ sở dữ liệu

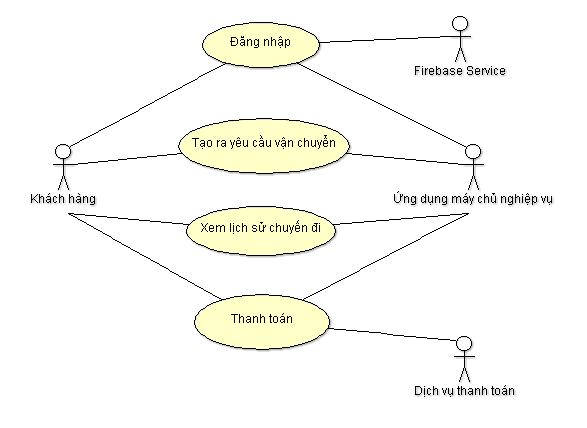
## 3.5 Phân tích và thiết kế Ứng dụng bắt xe của khách hàng

### 3.5.1 Chức năng

* Đăng nhập và đăng ký cho khách hàng sử dụng
* Tạo ra yêu cầu vận chuyển và gửi yêu cầu đến máy chủ nghiệp vụ
* Duyệt lịch sử chuyến đi gần đây của khách hàng
* Thanh toán bằng tiền mặt, thẻ debit/credit và lưu thông tin tài khoản thanh toán của khách hàng

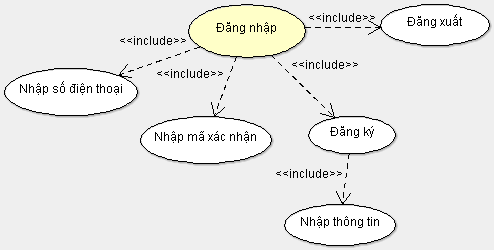
### 3.5.2 Các đối tượng tham gia

* Khách hàng sử dụng dịch vụ bắt xe trực tuyến
* Dịch vụ thanh toán qua thẻ
* Ứng dụng máy chủ xử lý nghiệp vụ
* Dịch vụ xác thực người dùng Firebase



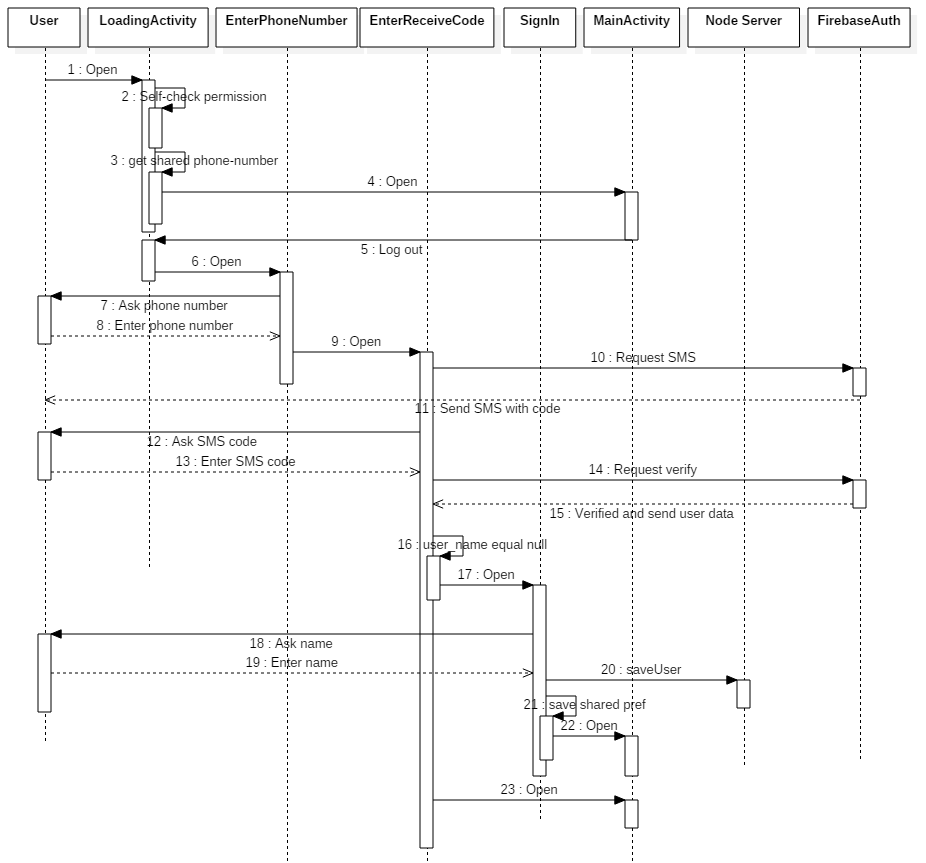
Hình ‑ Sơ đồ use-cases ứng dụng bắt xe của khách hàng

### 3.5.3 Thiết kế chức năng đăng nhập



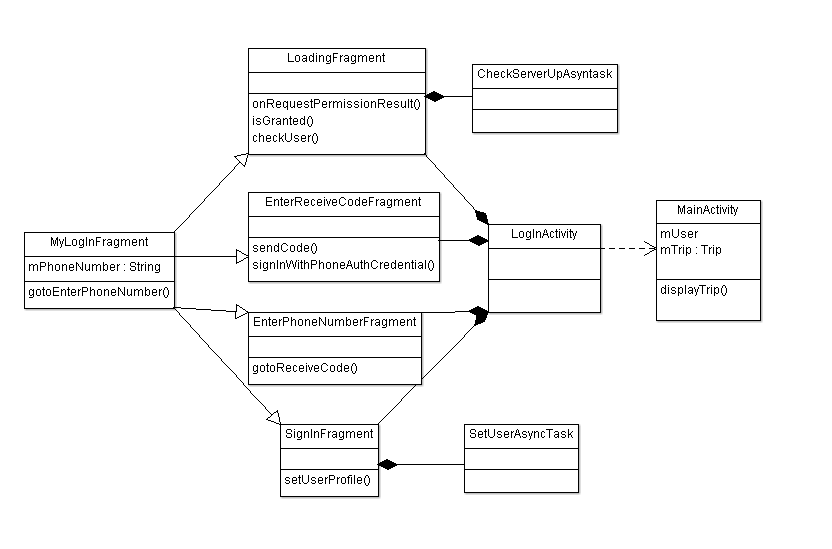
Hình ‑ Sơ đồ phân rã chức năng đăng nhập

* Đặc tả use-case:
  + Use-case đăng nhập cho phép khách hàng đăng nhập vào ứng dụng. Nếu khách hàng đã đăng ký tài khoản rồi thì yêu cầu khách hàng nhập số điện thoại để xác thực bằng mã số gửi về tin nhắn
  + Use-case đăng ký: nếu khách hàng chưa có tài khoản thì tạo tài khoản mới với họ tên và số điện thoại, sau đó được xác nhận thông qua mã số gửi về tin nhắn
  + Use‑case đăng xuất cho phép khách hàng đăng xuất tài khoản khỏi thiết bị để có thể đăng nhập bằng tài khoản khác
* Sơ đồ tuần tự quá trình đăng nhập của người dùng:



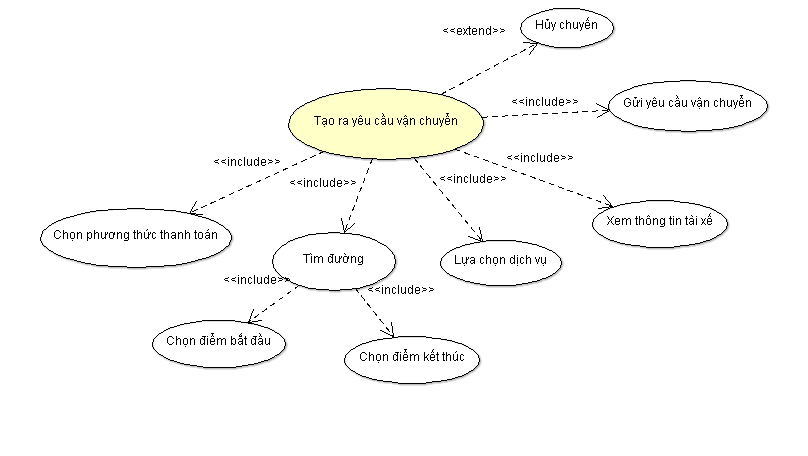
Hình ‑ Sơ đồ tuần tự của chức năng đăng nhập

* Sơ đồ lớp chức năng đăng nhập, các lớp được viết trên Android bằng ngôn ngữ lập trình Java:



Hình ‑ Sơ đồ lớp chức năng đăng nhập

### 3.5.4 Thiết kế chức năng tạo yêu cầu vận chuyển

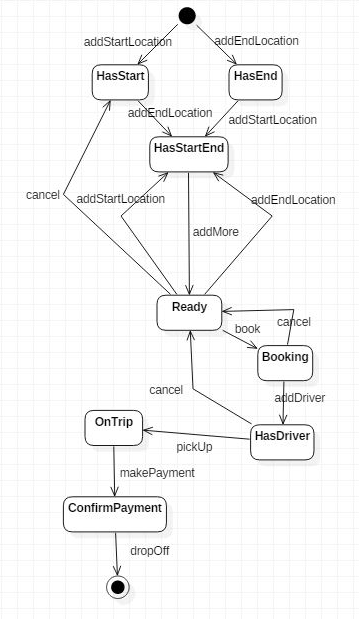


Hình ‑ Sơ đồ Use-case tạo yêu cầu vận chuyển

* Đặc tả Use-case:
  + Chức năng tạo yêu cầu vận chuyển cho phép khách hàng tìm đường đi; chọn loại dịch vụ; xem thông tin về giá cước, đường đi, tài xế; chọn phương thức thanh toán và hủy chuyến
  + Chức năng tìm đường cho phép khách hàng tìm điểm xuất phát và điểm kết thúc, nếu khách hàng không chọn điểm xuất phát thì điểm xuất phát mặc định sẽ là vị trí hiện tại của khách hàng
  + Chức năng lựa chọn dịch vụ cho phép khách hàng lựa chọn các loại dịch vụ, ví dụ như xe máy, taxi...
  + Chức năng chọn phương thức thanh toán cho phép khách hàng chọn thanh toán bằng tiền mặt hoặc thanh toán qua thẻ ngân hàng
  + Sau khi các thông tin về đường đi đã được xác nhận, khách hàng có thể gửi yêu cầu tìm tài xế hoặc hủy chuyến qua chức năng gửi yêu cầu vận chuyển và hủy chuyến
  + Sau khi server tìm được tài xế phù hợp, khách hàng sẽ được xem các thông tin về tài xế như họ tên, hình ảnh và số điện thoại qua chức năng xem thông tin tài xế
  + Chức năng hủy chuyến cũng cho phép khách hàng hủy chuyến đi trước khi tài xế chọn đón khách
* Chức năng tạo yêu cầu vận chuyển được thiết kế dựa trên trạng thái của chuyến đi, sơ đồ trạng thái của chuyến đi được mô tả ở hình 4-5 bên dưới.

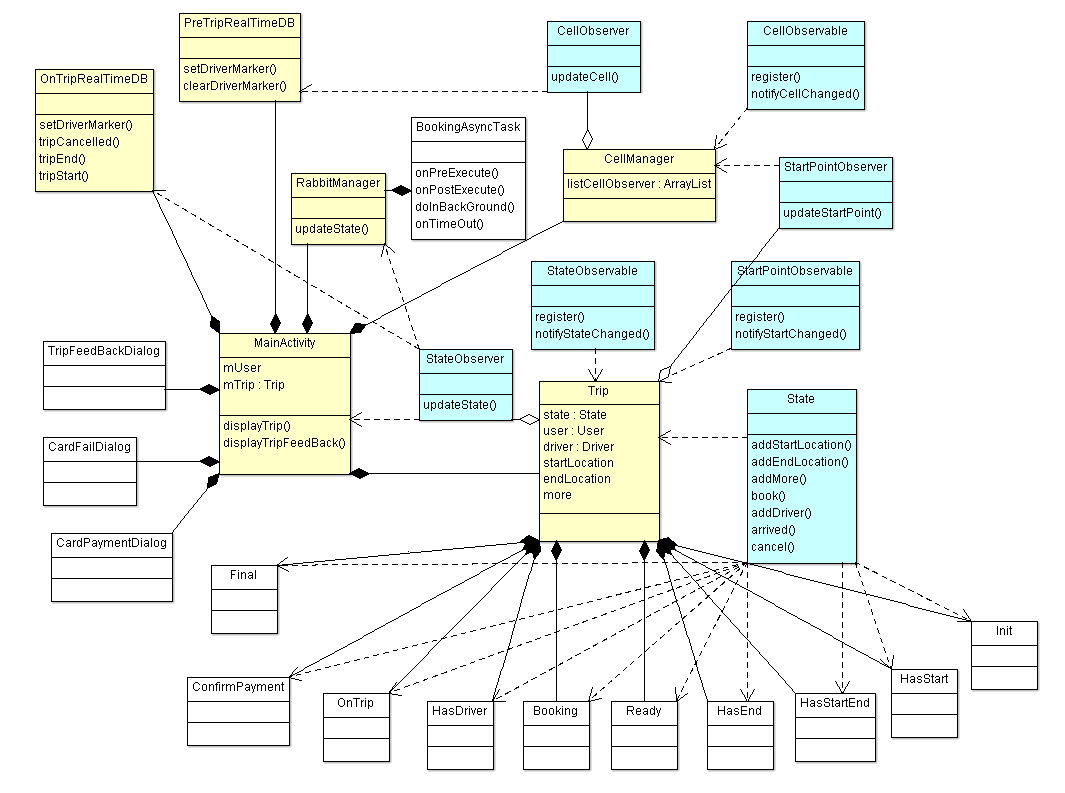
Giải thích sơ đồ trạng thái của chuyến đi:

* + Trạng thái khởi đầu (InitState) là trạng thái khách hàng vừa mở ứng dụng hoặc vừa đăng nhập vào ứng dụng hoặc vừa kết thúc chuyến đi, từ trạng thái này có thể thêm điểm xuất phát hoặc điểm đến để chuyển qua trạng thái tiếp theo
  + Trạng thái có điểm xuất phát (HasStart) là trạng thái khách hàng đã chọn được điểm xuất phát, điểm xuất phát có thể là vị trí hiện tại của khách hàng hoặc một vị trí tùy chọn. Từ trạng thái này có thể thêm điểm đến để sang trạng thái tiếp theo, hoặc thêm điểm xuất phát để truyển sang chính trạng thái này
  + Trạng thái có điểm kết thúc (HasEnd) là trạng thái khi khách hàng đã xác định được điểm đến, từ trạng thái này nếu khách hàng thêm điểm xuất phát thì chuyến đi sẽ chuyển qua trạng thái tiếp theo
  + Trạng thái có điểm đi và điểm đến (HasStartEnd) là trạng thái khi khách hàng đã tìm được điểm đi và điểm đến theo ý muốn, khi bắt đầu trạng thái này, ứng dụng sẽ gửi một yêu cầu đến server để tìm thông tin về đường đi, giá cước...
  + Nếu đường đi ở trạng thái trên là hợp lệ và server gửi các thông tin về đường đi, giá cước thì chuyến đi sẽ chuyển qua trạng thái tiếp theo là trạng thái sẵn sàng gọi chuyến (Ready). Ở trạng thái sẵn sàng, nếu khách hàng thay đổi điểm đi hoặc điểm đến thì chuyến đi sẽ chuyển về trạng thái trước, nếu khách hàng chấp nhận đặt chuyến thì ứng dụng sẽ gửi yêu cầu đặt chuyến đến server để server tìm kiếm tài xế phù hợp, khi đó chuyến đi sẽ chuyển sang trạng thái đang đặt chuyến (Booking)
  + Ở trạng thái đang đặt chuyến, nếu server tìm được tài xế phù hợp trong khoảng thời gian định trước thì chuyến đi sẽ chuyển qua trạng thái tiếp theo là trạng thái có tài xế (HasDriver), nếu quá thời gian định trước thì chuyến đi sẽ chuyển về trạng thái trước
  + Ở trạng thái có tài xế, nếu khách hàng hoặc tài xế hủy chuyến thì chuyến đi sẽ chuyển về trạng thái, nếu tài xế đã gặp khách hàng và chọn bắt đầu chuyến đi (pick up) thì chuyến đi sẽ chuyển qua trạng thái tiếp theo là trạng thái trong chuyến (OnTrip)
  + Ở trạng thái trong chuyến, nếu tài xế trả khách thì chuyến đi sẽ chuyển qua trạng thái chờ thanh toán
  + Trạng thái chờ thanh toán là trạng thái khách hàng đang trả tiền, có thể bằng tiền mặt hoặc bằng thẻ ngân hàng, sau khi xác nhận thanh toán, tài xế sẽ trả khách và kết thúc chuyến đi
  + Ở trạng thái kết thúc chuyến đi, khách hàng sẽ được yêu cầu nhận xét về chuyến đi và đánh giá tài xế, sau khi đánh giá, chuyến đi sẽ trở về trạng thái ban đầu và khách hàng có thể yêu cầu chuyến đi khác



Hình ‑ Trạng thái của chuyến đi trong quá trình tạo và thực hiện yêu cầu vận chuyển

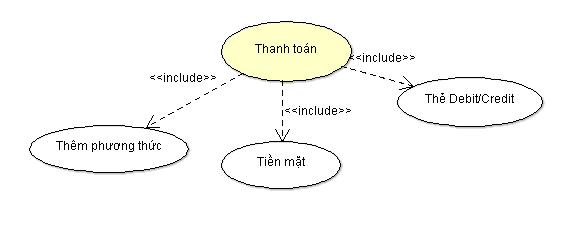
* Sơ đồ lớp chức năng tạo và thực hiện yêu cầu vận chuyển
* Lớp Trip lưu thông tin về chuyến đi bao gồm trạng thái chuyến đi, thông tin người dùng, tài xế, điểm xuất phát, điểm kết thúc, quãng đường, giá cước, loại dịch vụ... Lớp Trip được thiết kế theo mô hình publish-subcribe, khi điểm xuất phát hoặc trạng thái của chuyến đi thay đổi, lớp Trip sẽ thông báo cho các observer tương ứng biết được sự thay đổi đó
* Lớp CellManager quản lý về sự thay đổi của cell, lớp này lắng nghe đến sự thay đổi của điểm xuất phát trong lớp Trip, khi điểm xuất phát thay đổi, lớp CellManager sẽ kiểm tra xem cell mà người dùng muốn xuất phát có thay đổi hay không, nếu cell thay đổi, lớp này sẽ thông báo đến các observer tương ứng



Hình ‑ Mối quan hệ giữa chuyến đi và các đối tượng khác trong chương trình

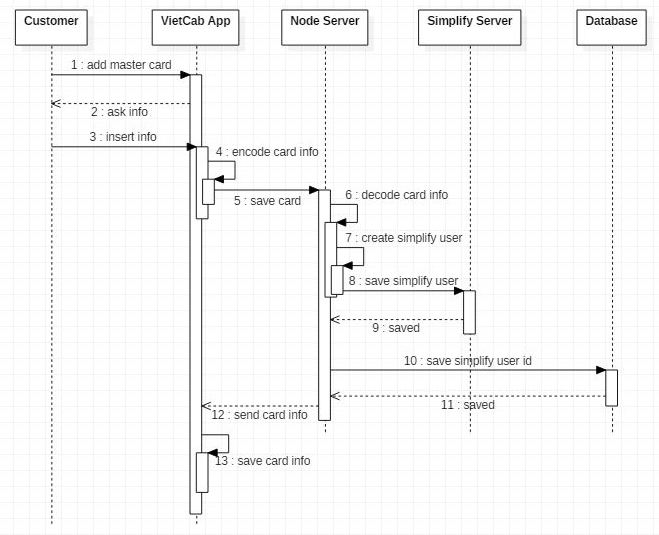
* Lớp PreTripRealTimeDB chịu trách nhiệm lắng nghe sự thay đổi của cell ở lớp CellManager sau đó hiển thị các tài xế khả dụng ở cell tương ứng theo thời gian thực. Thông tin theo thời gian thực của tài xế bao gồm loại dịch vụ và vị trí trên bản đồ. Thông tin của lớp này chỉ hiển thị khi khách hàng chưa tìm tài xế.
* Lớp RabbitManager lắng nghe đến sự thay đổi trạng thái (state) của lớp Trip, khi trạng thái của lớp Trip là Booking, lớp này sẽ gửi yều cầu nhận chuyến đến hàng đợi RabbitMQ. Nếu hàng đợi tìm được tài xế hợp lý, lớp RabbitManager sẽ có trách nhiệm cập nhật trạng thái của chuyến đi, nếu quá thời gian tìm kiếm mà hàng đợi không tìm được tài xế thì lớp này cũng sẽ có trách nhiệm cập nhật trạng thái của chuyến đi đó.
* Lớp OnTripRealTimeDB trịu trách nhiệm lắng nghe đến sự thay đổi trạng thái (state) của lớp Trip, khi trạng thái của lớp Trip là có tài xế (HasDriver) thì lớp này sẽ chỉ cập nhật vị trí của tài xế đã nhận chuyến. Đồng thời lớp này chịu trách nhiệm cập nhật các sự kiện mà tài xế phát sinh ra như hủy chuyến, đón khách và trả khách, khi đó lớp này sẽ cập nhật trạng thái tương ứng của chuyến đi

### 3.5.5 Thiết kế chức năng thanh toán



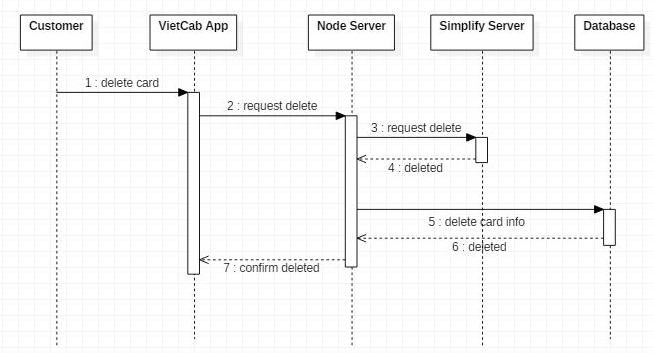
Hình ‑ Sơ đồ use-case chức năng thanh toán

* Đặc tả use-case:
* Use-case thanh toán cho phép khách hàng thêm phương thức thanh toán bằng thẻ tín dụng hoặc thẻ ghi nợ, ngoài ra use-case này còn cho phép khách hàng chọn phương thức thanh toán trước khi tìm chuyến đi như tiền mặt hay thẻ ngân hàng
* Chức năng thêm phương thức thanh toán (thẻ tín dụng hoặc thẻ ghi nợ):



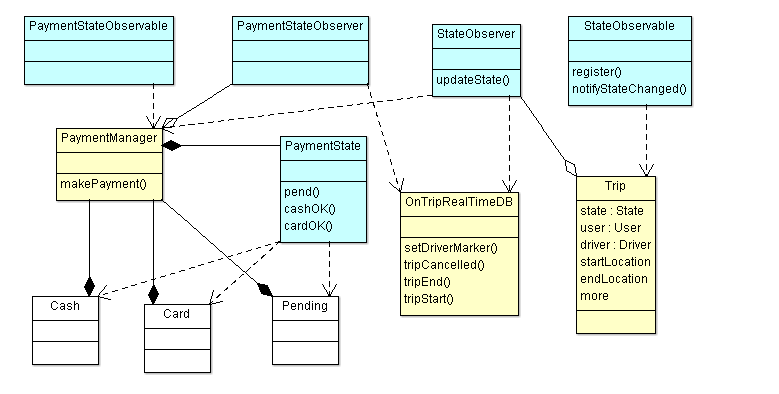
Hình ‑ Sơ đồ tuần tự chức năng thêm thẻ thanh toán

* Quá trình thêm thẻ thanh toán được mô tả bởi sơ đồ tuần tự hình 4-9. Khi khách hàng yêu cầu thêm thẻ thanh toán vào tài khoản, ứng dụng sẽ yêu cầu khách hàng nhập thông tin về mã số thẻ, thời gian hết hạn và mã bảo mật
* Nếu thông tin của khách hàng nhập vào là hợp lệ, thẻ của khách hàng sẽ được liên kết với tài khoản trên hệ thống bắt xe trực tuyến và được lưu trong cơ sở dữ liệu



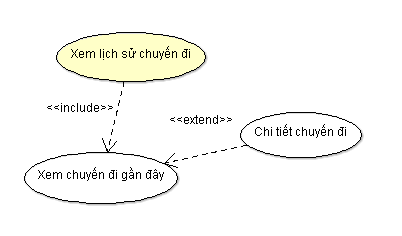
Hình ‑ Sơ đồ tuần tự quá trình xóa thẻ thanh toán

* Chức năng thêm phương thức thanh toán còn cho phép khách hàng xóa liên kết tài khoản của khách hàng trên hệ thống bắt xe trực tuyến với thẻ thanh toán, quá trình xóa thẻ được mô tả trên hình 4-10 ở trên.
* Chức năng thanh toán chuyến đi
* Lớp PaymentManager chịu trách nhiệm lắng nghe đến sự thay đổi trạng thái của chuyến đi (lớp Trip), nếu trạng thái của lớp Trip là ConfirmPayment thì lớp này sẽ yêu cầu thanh toán tùy vào lựa chọn ban đầu của khách hàng. Nếu khách hàng thanh toán qua thẻ, lớp này sẽ gửi yêu cầu thanh toán đến server, nếu server gửi xác nhận thanh toán thành công thì lớp này chịu trách nhiệm thông báo thanh toán thành công cho tài xế. Nếu thanh toán bằng thẻ không thành công hoặc khách hàng chọn thanh toán bằng tiền mặt thì lớp này sẽ thông báo đến tài xế để tài xế thu tiền mặt



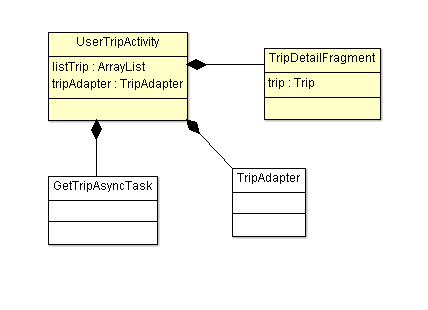
Hình ‑ Sơ đồ lớp chức năng thanh toán chuyến đi

### 3.5.6 Thiết kế chức năng xem lịch sử chuyến đi



Hình ‑ Sơ đồ Use-case chức năng xem lịch sử chuyến đi

* Đặc tả use-case:
* Use-case xem lịch sử chuyến đi cho phép người dùng xem các chuyến đi gần đây và chi tiết mỗi chuyến đi bao gồm: thông tin chuyến đi như điểm xuất phát, điểm dừng, quãng đường, giá tiền, thời gian và trạng thái (thành công hoặc bị hủy); thông tin về tài xế bao gồm họ tên, hình ảnh.
* Sơ đồ lớp:



Hình ‑ Sơ đồ lớp chức năng xem lịch sử chuyến đi

* Khi khách hàng lựa chọn xem lịch sử chuyến đi, lớp GetTripAsyncTask sẽ gửi yêu cầu xem lịch sử chuyến đi đến server, phía server sẽ dựa vào id của khách hàng để truy vấn trong cơ sở dữ liệu và trả về kết quả cho ứng dụng dưới dạng tệp tin JSON. Sau khi nhận được tập tin JSON, ứng dụng của khách hàng sẽ phân tích dữ liệu và hiển thị danh sách các chuyến đi gần đây.
* Khi khách hàng lựa chọn một chuyển đi (không bị hủy) thì thông tin chi tiết về chuyến đi sẽ được hiển thị bao gồm các thông tin đã mô tả trong đặc tả use-case.

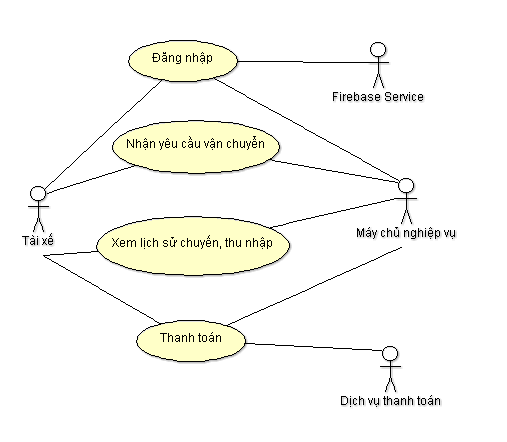
## 3.6 Phân tích và thiết kế ứng dụng nhận chuyến xe của tài xế

### 3.6.1 Chức năng

* Đăng nhập dành cho tài xế
* Nhận yêu cầu vận chuyển
* Xem thu nhập và lịch sử chuyến đi
* Thanh toán

### 3.6.2 Các đối tượng tham gia

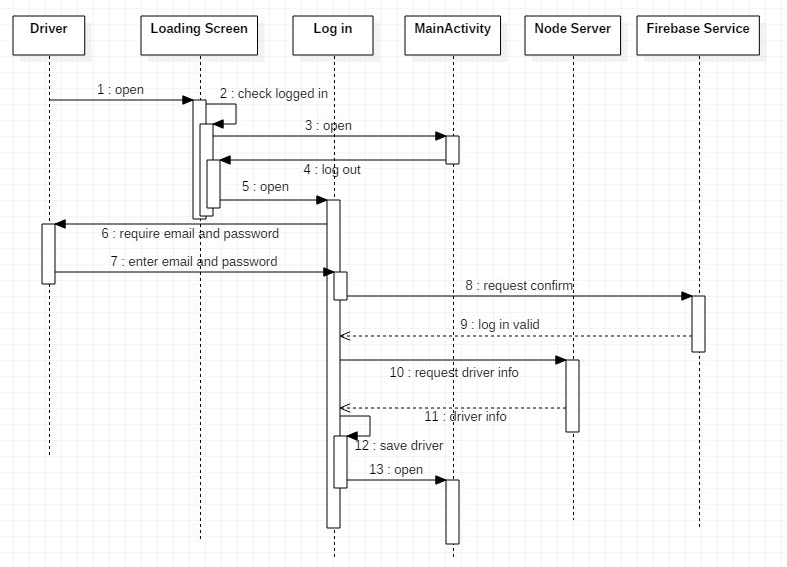
* Tài xế lái xe
* Ứng dụng máy chủ nghiệp vụ
* Dịch vụ thanh toán
* Dịch vụ xác thực Firebase



Hình ‑ Sơ đồ use-cases ứng dụng nhận chuyến xe của tài xế

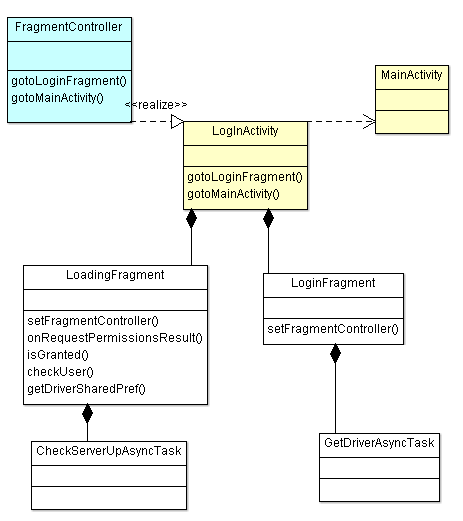
### 3.6.3 Thiết kế chức năng đăng nhập

* Đặc tả use-cases:
* Các đối tác muốn tham gia làm tài xế của hệ thống bắt xe trực tuyến sẽ phải đăng ký trên trang chủ của hệ thống, sau khi được admin phê duyệt hồ sơ, tài xế sẽ được chấp nhận (hoặc loại bỏ) tham gia vào hệ thống
* Chức năng đăng nhập trên ứng dụng nhận chuyến của tài xế cho phép tài xế nhập vào email và mật khẩu xác thực để đăng nhập vào hệ thống
* Sơ đồ tuần tự quá trình đăng nhập vào hệ thống được mô tả ở hình dưới:



Hình ‑ Sơ đồ tuần tự quá trình đăng nhập của tài xế

* Sơ đồ lớp chức năng đăng nhập được mô tả bởi hình 3-23 bên dưới
* Lớp LoadingFragment có tác dụng kiểm tra xem tài xế đã đăng nhập chưa, nếu đăng nhập rồi thì chuyển hướng đến MainActivity, nếu không sẽ chuyển hướng đến LoginFragment. Ngoài ra lớp này cũng có tác dụng kiểm tra sự cho phép của người dùng về quyền truy cập vị trí và tình trạng kết nối đến máy chủ.
* Lớp LoginFragment có chức năng cho phép người dùng nhập thông tin email và mật khẩu và xác nhận nếu thông tin khớp thì sẽ chuyến hướng đến MainActivity
* Khi tài xế mở ứng dụng, lớp LoginActivity sẽ được kích hoạt, lớp này có tác dụng chứa hai Fragment trên
* Lớp CheckServerUpAsyncTask có tác dụng kiểm tra kết nối đến máy chủ
* Lớp GetDriverAsyncTask có tác dụng lấy dữ liệu thông tin về tài xế khi tài xế đã nhập đúng email và mật khẩu



Hình ‑ Sơ đồ lớp chức năng đăng nhập

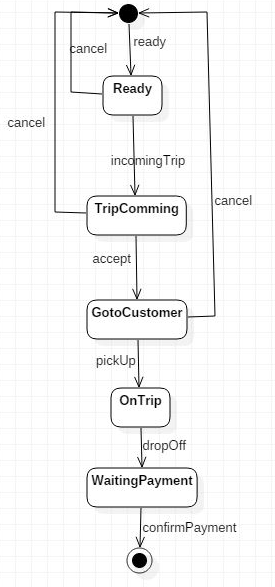
### 3.6.4 Thiết kế chức năng nhận yêu cầu vận chuyển

* Sơ đồ use-case phân rã chức năng:
* Chức năng nhận yêu cầu vận chuyển cho phép tài xế xem thông tin về chuyến đi mỗi khi có chuyến đi được gửi đến, từ chối chuyến đi nếu muốn hoặc chấp nhận chuyến đi
* Chức năng xem thông tin chuyến đi cho phép tài xế biết thông tin về khách hàng (họ tên, số điện thoại) và thông tin về chuyến đi (điểm đón khách, điểm trả khách, giá cước, khoảng cách đường đi...)
* Chức năng từ chối chuyến đi cho phép tài xế không nhận chuyến đi nếu muốn



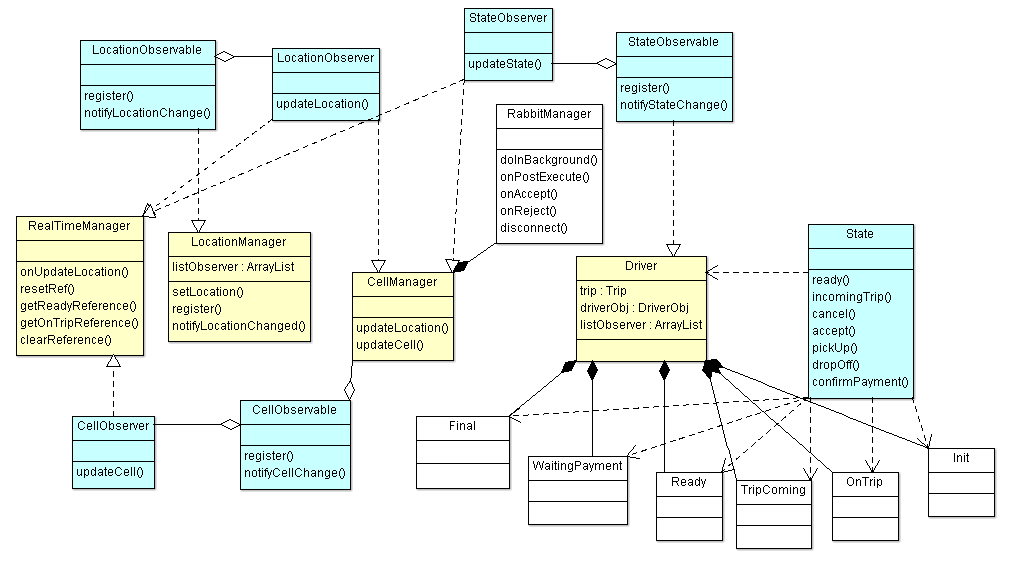
Hình ‑ Sơ đồ phân rã chức năng nhận yêu cầu vận chuyển

* Chức năng chấp nhận chuyến đi cho phép tài xế nhận chuyến, sau khi nhận chuyển tài xế phải đón khách và trả khách, tuy nhiên tài xế vẫn có thể hủy chuyến nếu chuyến đi chưa bắt đầu
* Chức năng cập nhật trạng thái của tài xế cho phép ứng dụng nhận chuyến xe cập nhật tọa độ của tài xế và trạng thái của tài xế theo thời gian thực
* Chức năng nhận yêu cầu vận chuyển được thiết kế dựa trên trạng thái của tài xế, trạng thái của tài xế được mô tả dưới hình 5-5 sau:
* Trạng thái khởi đầu (InitState) là trạng thái tài xế vừa đăng nhập vào ứng dụng hoặc vừa hoàn thành (hoặc hủy) một chuyến đi. Từ trạng thái này tài xế có thể chuyến sang trạng thái sẵn sàng
* Trạng thái sẵn sàng là trạng thái tài xế có thể nhận chuyến đi, từ trạng thái này nếu có chuyến đi từ server gửi đến, tài xế sẽ nhận được thông báo có chuyến đi và chuyển qua trạng thái TripComing (có chuyến đi được gửi đến chờ xác nhận)
* Từ trạng thái TripComing, nếu tài xế không chấp nhận chuyến đi thì trạng thái của tài xế sẽ chuyển về trạng thái khởi đầu, nếu tài xế chấp nhận chuyến đi thì chuyển qua trạng thái tiếp theo là GotoCustomer (đang đi đến vị trí đón khách)



Hình ‑ Trạng thái của tài xế

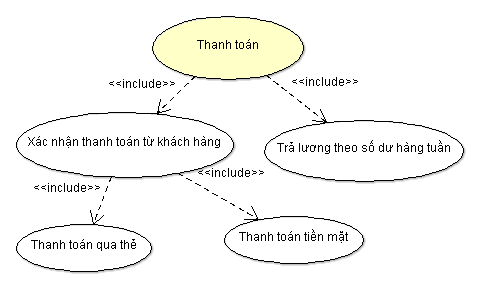
* Từ trạng thái GotoCustomer, nếu tài xế hủy chuyến, thì trạng thái của tài xế sẽ chuyển về trạng thái khởi đầu, nếu tài xế đón khách thì trạng thái của tài xế sẽ chuyển sang OnTrip (trong chuyến)
* Nếu tài xế đang trong trạng thái OnTrip và chọn trả khách thì tài xế sẽ chuyển qua trạng thái chờ thanh toán (WaitingPayment)
* Trạng thái chờ thanh toán sẽ chờ thông báo từ phía ứng dụng của khách hàng để biết phương thức thanh toán là tiền mặt hay thẻ, khi tài xế xác nhận đã thanh toán thì chuyến đi kết thúc và tài xế sẽ trở về trạng thái khởi đầu
* Sơ đồ lớp chức năng nhận yêu cầu vận chuyển:
* Lớp LocationManager chịu trách nhiệm thông báo về sự thay đổi vị trí của tài xế cho các lớp observer tương ứng mỗi khi vị trí của tài xế thay đổi
* Lớp CellManager lắng nghe đến sự kiện thay đổi vị trí và thay đổi trạng thái của tài xế, nếu vị trí thay đổi một khoảng cách định trước thì lớp này sẽ gửi yêu cầu đến server để lấy thông tin về cell hiện tại. Nếu cell thực sự thay đổi thì lớp này sẽ ngắt kết nối đến hàng đợi ở cell hiện tại và kết nối đến hàng đợi ở cell mới thông qua lớp RabbitManager. Ngoài ra lớp này sẽ tùy thuộc vào trạng thái của tài xế mà cập nhật kết nối đến hàng đợi RabbitMQ, ví dụ khi tài xế chuyển sang trạng thái sẵn sàng (Ready) thì lớp này sẽ đăng ký hàng đợi RabbitMQ cùng với cell tương ứng.
* Lớp RealTimeManger chịu trách nhiệm cập nhật thông tin của tài xế lên cơ sở dữ liệu thời gian thực. Thông tin này bao gồm: trạng thái của tài xế, vị trí hiện tại (tọa độ theo Latitude và Longitude), họ tên, loại dịch vụ. Tuy nhiên lớp này sẽ chỉ cập nhật khi trạng thái của tài xế khác trạng thái khởi đầu.
* Lớp Driver sẽ lưu thông tin của tài xế và chuyến đi, đồng thời có trách nhiệm thông báo cho các lớp observer tương ứng về sự thay đổi trạng thái của tài xế



Hình ‑ Sơ đồ lớp chức năng nhận yêu cầu vận chuyển

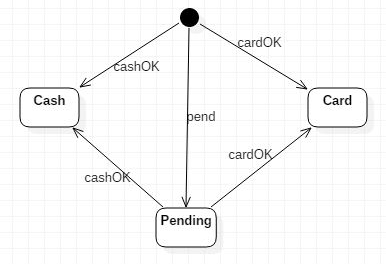
### 3.6.5 Thiết kế chức năng thanh toán

* Sơ đồ Use-case phân rã chức năng:
* Chức năng thanh toán bao gồm hai chức năng chính là thanh toán khi kết thúc chuyến đi và thanh toán trả lương cho tài xế
* Use-case xác nhận thanh toán từ khách hàng thông báo cho tài xế biết phương thức thanh toán cho chuyến đi khi kết thúc chuyến đi, nếu khách hàng thanh toán bằng tiền mặt thì tài xế sẽ nhận được thông báo thu tiền mặt, nếu khách hàng thanh toán bằng thẻ ngân hàng thành công thì tài xế cũng nhận được thông báo xác nhận giao dịch qua thẻ đã thành công
* Use-case trả lương theo số dư hàng tuần cho phép hệ thống trả lương cho tài xế dựa trên số dư tương ứng trong tài khoản ảo, chức năng này cần liên kết và hợp tác với các ngân hàng nội địa nên tạm thời chưa thiết kế



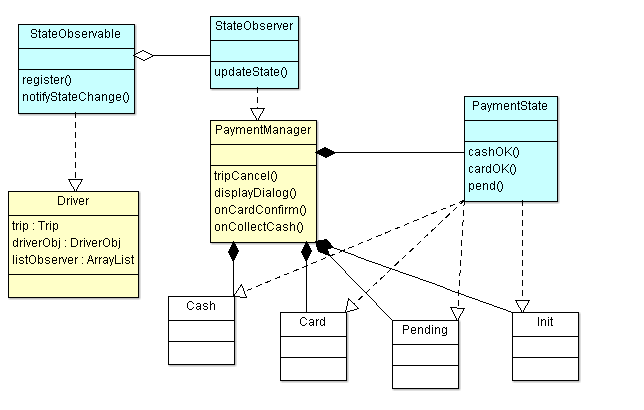
Hình ‑ Sơ đồ phân rã chức năng thanh toán

* Chức năng xác nhận thanh toán từ khách hàng thiết kế theo trạng thái của quá trình thanh toán, trạng thái quá trình thanh toán được mô tả ở sơ đồ trạng thái dưới đây
* Trạng thái khởi đầu là trạng thái chưa thanh toán, từ trạng thái này nếu khách hàng chọn thanh toán bằng tiền mặt thì lập tức trạng thái thanh toán sẽ chuyển sang Cash (tiền mặt), nếu khách hàng chọn thanh toán bằng thẻ thì sẽ phải chờ một thời gian để server kiểm tra tình trạng thanh toán, quá trình thanh toán chuyển sang trạng thái Pending
* Ở trạng thái Pending, nếu server phản hồi rằng thanh toán thành công thì quá trình thanh toán chuyển qua trạng thái Card (thẻ), nếu không thành công thì trạng thái thanh toán chuyển qua Cash.



Hình ‑ Trạng thái quá trình thanh toán sau chuyến đi

* Sơ đồ lớp chức năng thanh toán sau chuyến đi
* Lớp Driver như mô tả ở phần 3-29, sẽ lưu thông tin về chuyến đi và thông báo sự thay đổi trạng thái của tài xế đến các observer tương ứng
* Lớp PaymentManager sẽ lắng nghe đến sự thay đổi trạng thái của chuyến đi, nếu trạng thái của tài xế là GotoCustomer thì lớp này sẽ lắng nghe đến cơ sở dữ liệu thời gian thực để cập nhật trạng thái thanh toán từ ứng dụng của khách hàng. Nếu trạng thái của tài xế là Final (kết thúc) thì lớp này sẽ ngắt lắng nghe đến cơ sở dữ liệu thời gian thực để tránh lỗi có thể xảy ra



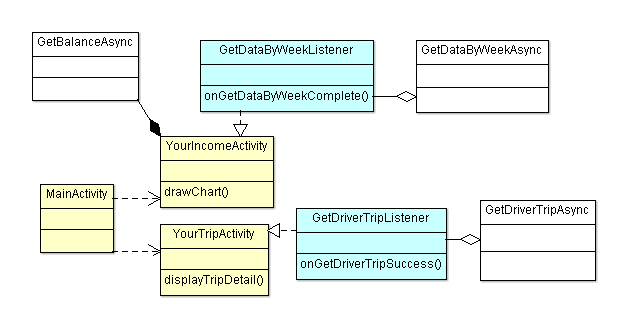
Hình ‑ Sơ đồ lớp chức năng thanh toán sau chuyến đi

### 3.6.6 Thiết kế chức năng xem lịch sử và xem thu nhập



Hình ‑ Sơ đồ phân rã chức năng xem lịch sử và thu nhập

* Đặc tả use-case:
* Use-case xem lịch sử chuyến đi cho phép tài xế xem các chuyến đi gần đây và chi tiết mỗi chuyến đi. Thông tin về chuyến đi bao gồm: thời gian, tổng tiền, điểm đi, điểm đến, khoảng cách, phương thức thanh toán, nhận xét và đánh giá của khách hàng.
* Use-case xem thu nhập sẽ thống kê thu nhập hàng tuần (từ thứ 2 đến chủ nhật) bằng cách tính tổng thu nhập các ngày trong tuần và vẽ lên đồ thị dạng cột
* Sơ đồ lớp:



Hình ‑ Sơ đồ lớp chức năng xem lịch sử và xem thu nhập

* Lớp GetDataByWeekAsync sẽ gửi yêu cầu đến server để nhận về dữ liệu dạng JSON là thông tin về thu nhập của tài xế ở một số tuần gần đây
* Lớp YourIncomeActivity có trách nhiệm nhận dữ liệu thu nhập của tài xế từ lớp GetDataByWeekAsync, sau đó phân tích dữ liệu từ tệp JSON và vẽ lên đồ thị
* Lớp GetDriverTrip sẽ có trách nhiệm gửi yêu cầu đến server để nhận về dữ liệu dạng JSON là thông tin về một số chuyến đi gần đây của tài xế
* Lớp YourTripActivity có trách nhiệm phân tích dữ liệu từ tệp JSON sau đó hiển thị cho tài xế, đồng thời lớp này cũng hiển thị chi tiết chuyến đi nếu tài xế muốn xem
* Lớp GetBalanceAsync sẽ lấy thông tin về số tiền hiện tại trong tài khoản ảo của tài xế.

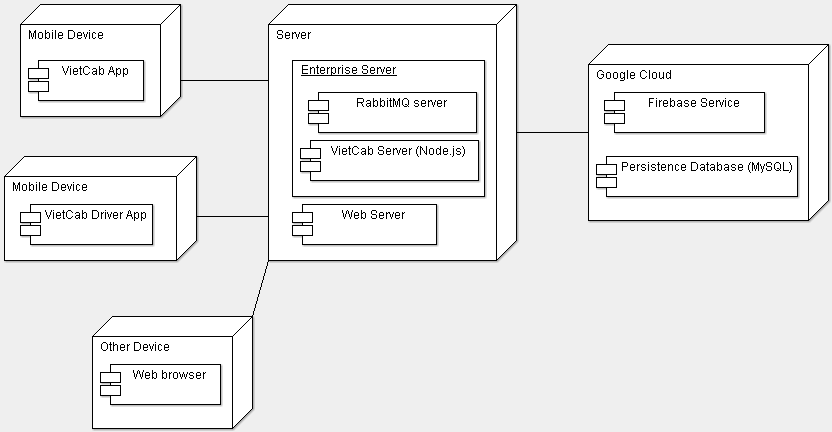
Kết luận:

Chương 3 dựa vào các khảo sát từ thực tế cùng các qui trình và lý thuyết cần thiết, tôi đã phân tích các yêu cầu của hệ thống, từ đó thiết kế ra một hệ thống bắt xe trực tuyến hoàn chỉnh, sẵn sàng để triển khai.

# Triển khai hệ thống

Chương 4 sẽ trình bày về việc triển khai hệ thống và các kết quả các ứng dụng đã được triển khai và chạy thử trên thiết bị thật

### 4.1 Sơ đồ triển khai của hệ thống

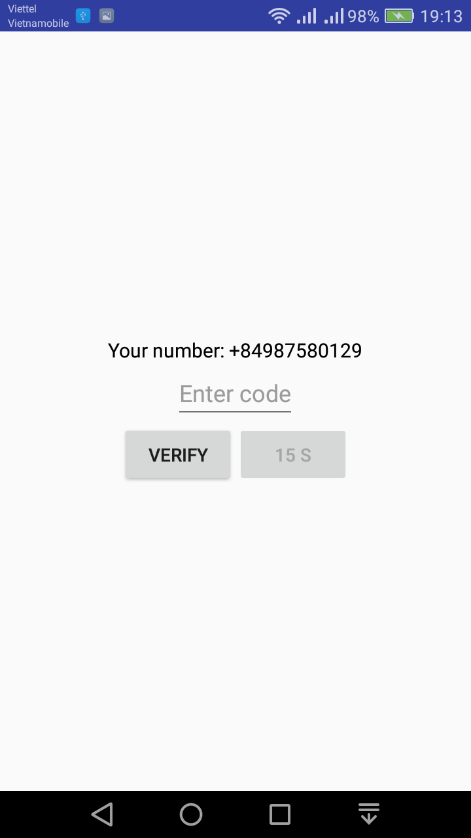


Hình 4-0 Sơ đồ triển khai hệ thống

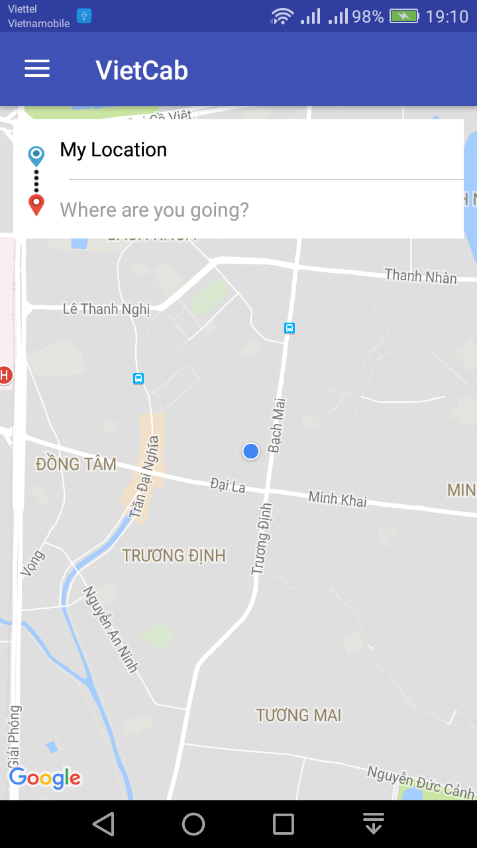
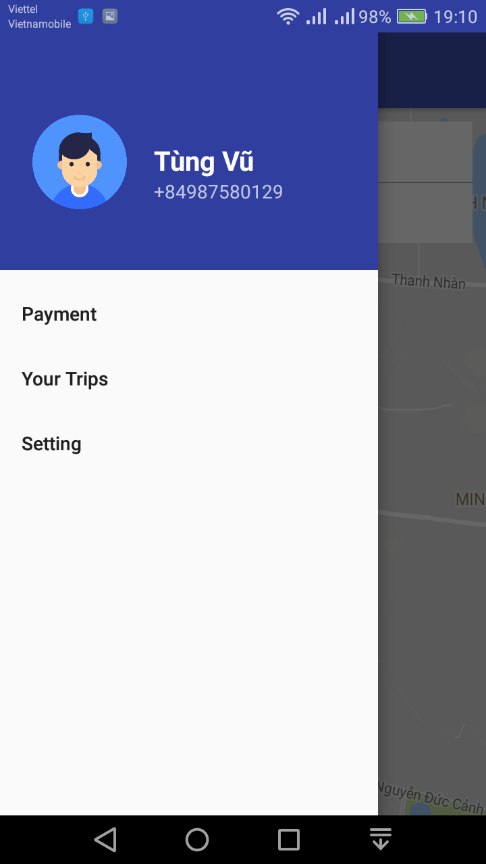
* + Máy chủ nghiệp vụ, hàng đợi RabbitMQ và máy chủ web sẽ được triển khai chung trên một máy chủ.
  + Cơ sở dữ liệu MySQL sẽ sử dụng dịch vụ của Google Cloud. Dịch vụ xác thực bằng tin nhắn sử dụng dịch vụ fire-base của Google.
  + Các thiết bị chạy ứng dụng nhận chuyến xe của tài xế, bắt xe của khách hàng và các thiết bị khác sẽ kết nối đến server qua giao thức TCP/IP.

## 4.2 Ứng dụng bắt xe của khách hàng

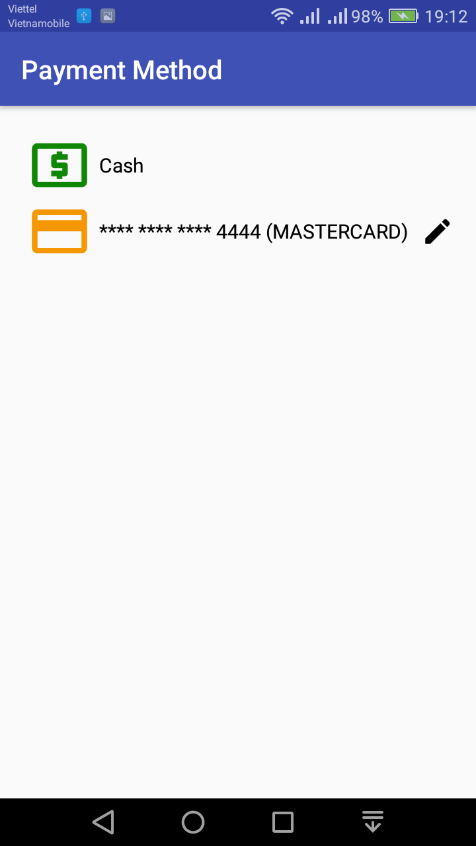
Ứng dụng bắt xe của khách hàng được lập trình bằng ngôn ngữ lập trình Java và được chạy trên các thiết bị Android từ phiên bản Android 5.0 trở lên, hình ảnh ứng dụng được đưa ra bên dưới

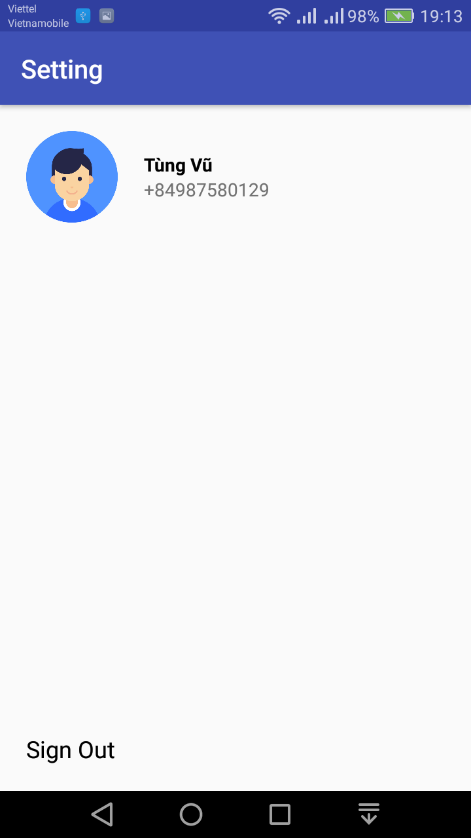
Hình ‑ Màn hình đăng nhập  Hình ‑ Màn hình xác thực bằng OTP

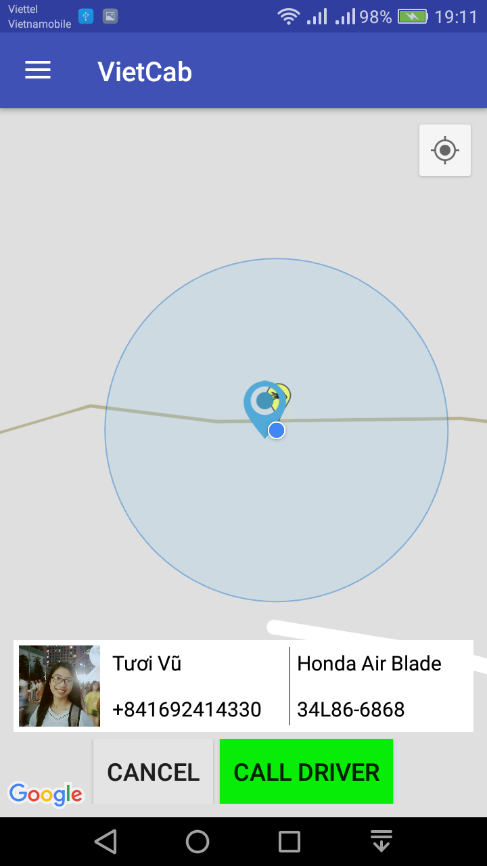
Hình ‑ Màn hình chính Hình ‑ Menu chức năng chính

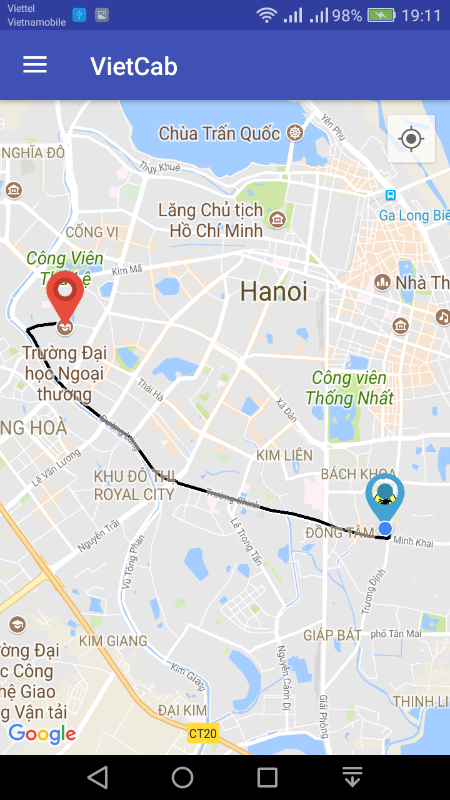
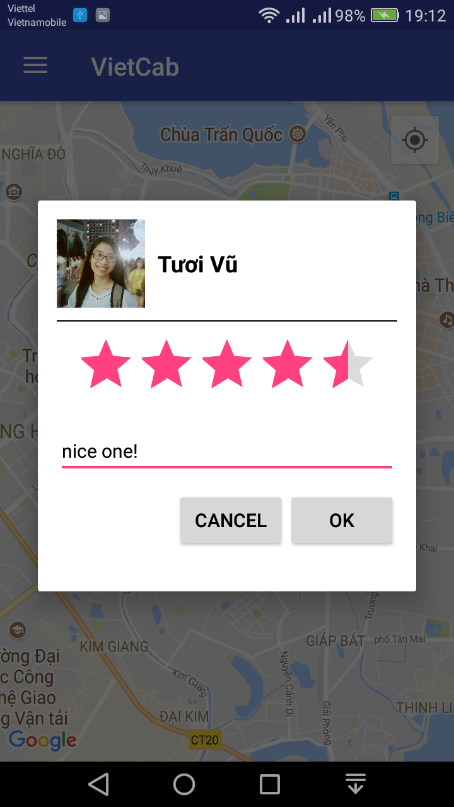
Hình ‑ Thêm thẻ thanh toán Hình ‑ Lịch sử chuyến đi

Hình ‑ Chi tiết chuyến đã đi Hình ‑ Màn hình đăng xuất

Hình ‑ Tạo chuyến đi mới Hình ‑ Tài xế chấp nhận chuyến đi

Hình ‑ Trong chuyến đi Hình ‑ Đánh giá tài xế



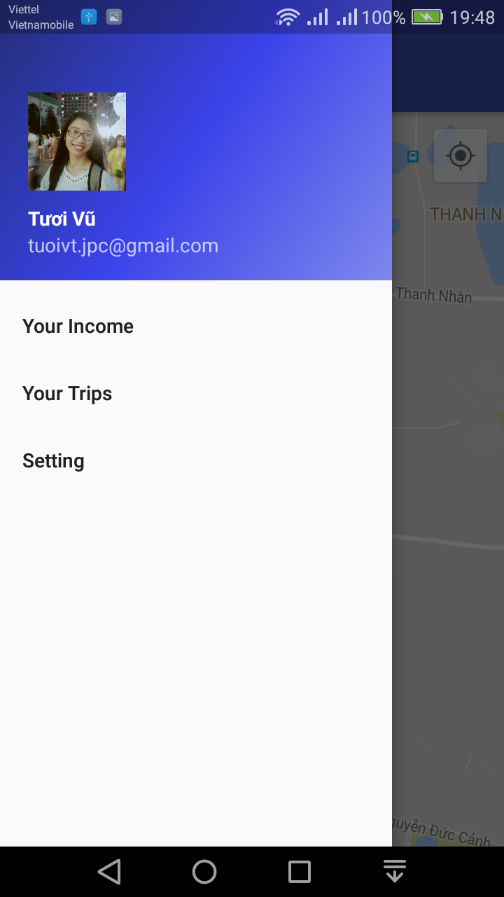
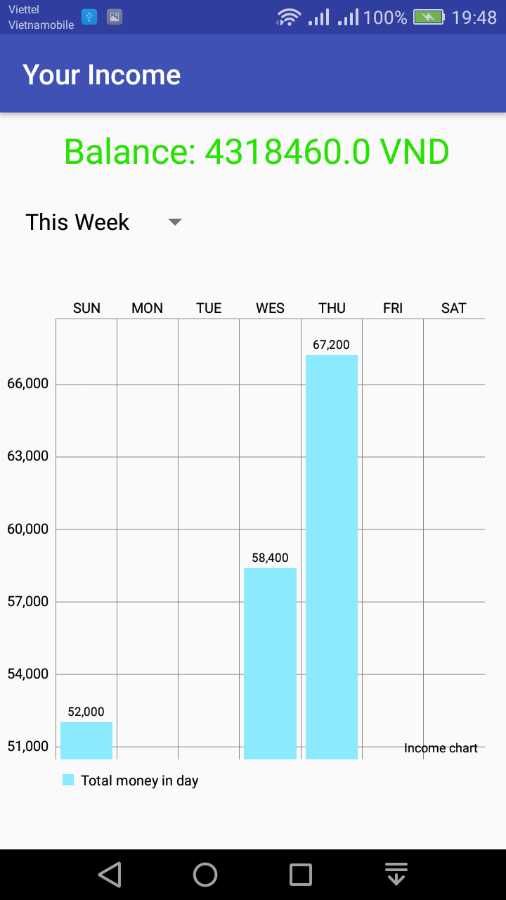
Hình ‑ Thanh toán qua thẻ thành công

## 4.3 Ứng dụng nhận chuyến xe của tài xế

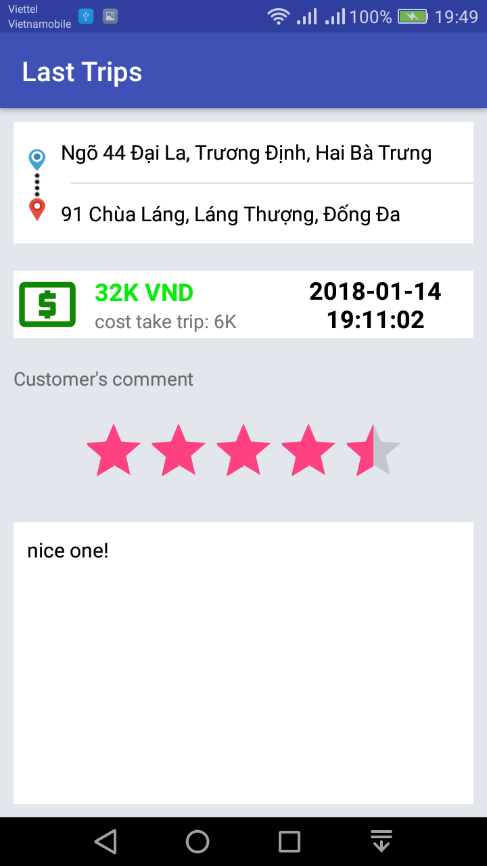
Ứng dụng nhận chuyến xe của tài xế được lập trình bằng ngôn ngữ lập trình Java và được chạy trên các thiết bị Android từ phiên bản Android 5.0 trở lên, hình ảnh ứng dụng được đưa ra bên dưới

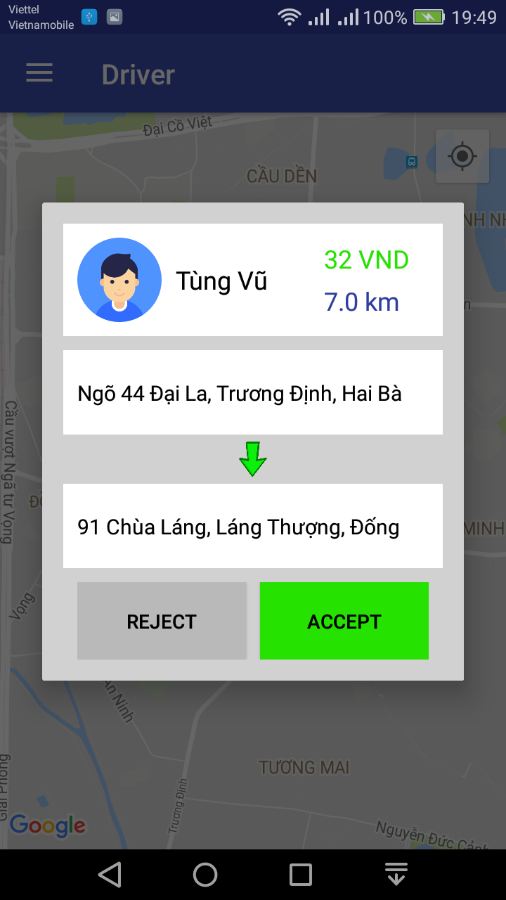
Hình ‑ Màn hình đăng nhập Hình ‑ Giao diện chính

Hình ‑ Menu chức năng chính Hình ‑ Thống kê thu nhập

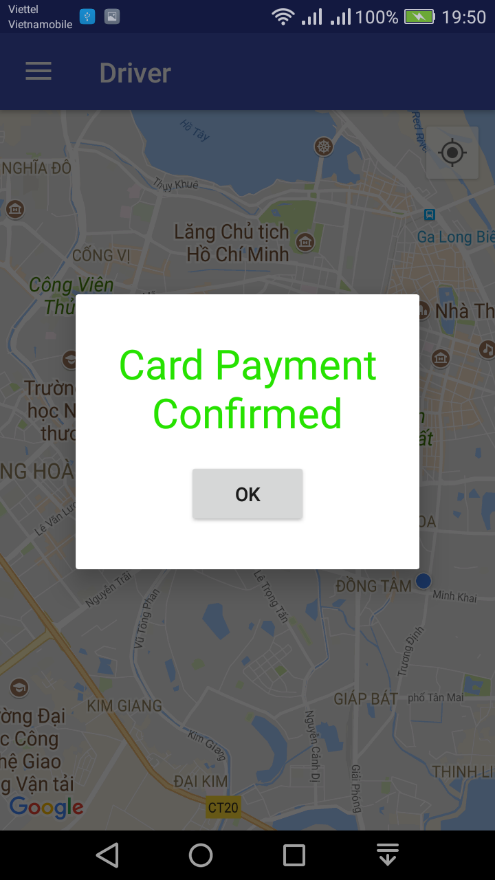
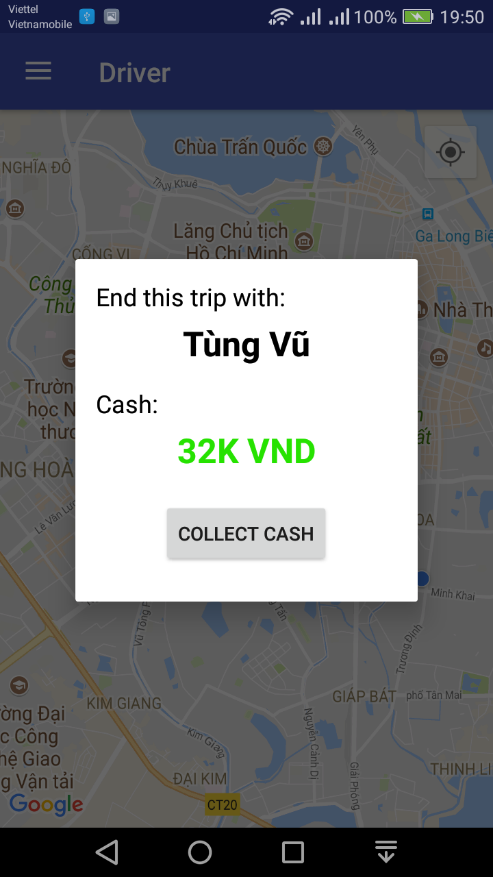
Hình ‑ Lịch sử chuyến đi  Hình ‑ Chi tiết chuyến đã đi

Hình ‑ Sắn sàng nhận chuyến  Hình ‑ Có chuyến đi được gửi đến

Hình ‑ Tài xế đến đón khách Hình ‑ Màn hình trong chuyến đi

Hình ‑ Thanh toán qua thẻ Hình ‑ Thanh toán bằng tiền mặt

Kết luận:

Trong chương 4 này, tôi đã trình bày được một số kết quả thực hiện dự án bằng việc triển khai các ứng dụng của khách hàng và của tài xế trên các thiết bị điện thoại Android. Dựa vào những gì đã làm được thì tôi đã đi đến kết luận và đưa ra những phương hướng phát triển cho đề tải ở chương cuối.

# Kết luận và hướng phát triển

## 5.1 Kết quả đạt được

Trong quá trình thực hiện đề tài, tôi đã hoàn thành mục tiêu đề ra và đạt được những kết quả sau:

* + Cải thiện kỹ năng phân tích thiết kế hệ thống
  + Nâng cao kỹ năng lập trình và triển khai hệ thống
  + Tìm hiểu các công nghệ, công cụ mới
  + Khảo sát, phân tích thiết kế hệ thống
  + Xây dựng các ứng dụng cho tài xế và khách hàng
  + Xây dựng ứng dụng máy chủ nghiệp vụ
  + Triển khai các ứng dụng

Bên cạnh những kết quả trên thì tôi cũng gặp phải những khó khăn về đánh giá hệ thống, các thiết bị triển khai, vấn đề về khảo sát đánh giá hệ thống đã có sẵn.

## 5.2 Phương hướng phát triển

Dựa vào những kết quả trên, tôi xin đề ra những định hướng phát triển trong tương lai của hệ thống bắt xe trực tuyến như sau:

* + Phát triển các ứng dụng cho điện thoại chạy hệ điều hành IOS dựa trên những phân tích thiết kế đã được thực hiện
  + Phát triển thêm các chức năng mới cho hệ thống như khuyến mại, quảng cáo, cá nhân hóa người dùng
  + Triển khai hệ thống để đưa tới tay người sử dụng
  + Cải thiện giao diện và trải nghiệm người dùng bằng việc thêm vào các hình ảnh hoạt họa, các chuyển động và âm thanh tương tác
  + Phát triển hệ thống phục vụ người dùng trên khắp thế giới

# Tài liệu tham khảo

[1] Bert Bates, Kathy Sierra, Eric Freeman, Elisabeht Robson, *Headfirst Design Pattern*, O’Reilly Media, 2009

[2] Martin Cagan, *How to write a Good PRD*, Sillicon Valley Product Group, 2005

[3] Gregor Hohpe, Bobby Woolf, *Enterprise Integration Patterns*, The Addison-Wesley Signature Series

[4] Evan M. Hahn, *Express in Action*, Manning Shelter Island, 2016

[5] Seyed M.M, “Saied” Tahaghoghi, Hugh E. Williams, *Learning MySQL,* O’Reilly Media, 2007

[6] Mark L. Murphy, *The Busy Coder’s Guide to Android Development*, Commons Ware, 2016

[7] Uber Engineering, *Cherami: Uber Enginreering’s Durable and Scalable Task Queue in Go*, <https://eng.uber.com/cherami/> , truy cập lần cuối 14/01/2018

[8] Enterprise Intergration Patterns, *Competing Consumer*,

<http://www.enterpriseintegrationpatterns.com/patterns/messaging/CompetingConsumers.html>, truy cập lần cuối 14/01/2018

[9] RabbitMQ Tutorial, *Work Queues*, https://www.rabbitmq.com/tutorials/tutorial-two-java.html, truy cập lần cuối 14/01/2018

[10] Firebase Documentation, <https://firebase.google.com/docs/?authuser=0>, truy cập lần cuối 14/01/2018