# **ARCHITECTURE HANDBOOK**

# **HỆ THỐNG ĐẶT XE TRỰC TUYẾN GRIB**

# **OOAD Nhóm 3**

| **21020403** | **Cao Tiến Thắng** |
| --- | --- |
| **21020785** | **Nguyễn Đăng Quang** |
| **21020422** | **Vũ Thị Thành Vinh** |
| **21020399** | **Nguyễn Vũ Thanh Tùng** |
| **21020787** | **Hoàng Văn Quyền** |

# 

[**ARCHITECTURE HANDBOOK 1**](#_uuc8tix4gndd)

[**HỆ THỐNG ĐẶT XE TRỰC TUYẾN GRIB 1**](#_67funbo1ex4h)

[**OOAD Nhóm 3 1**](#_p2pyg7nvo0hm)

[**1. Giới thiệu 3**](#_tne1csdp3rm7)

[1.1 Mục đích 3](#_3z7wdkpcve2q)

[1.2 Quy ước tài liệu 3](#_lmwu8w570dgc)

[1.3 Độc giả và cách đọc tài liệu 3](#_ygh20u9q6z29)

[1.4 Phạm vi dự án 4](#_c993jau2jclx)

[1.5 Tài liệu tham khảo 4](#_thdoy1qumkc5)

[**2. Các cơ chế kiến trúc 4**](#_qf9fnho89r8j)

[2.1 Các cơ chế phân tích 4](#_1dmdm3wc31jd)

[2.1.1 Cơ chế lưu trữ 4](#_u9yn9u3nio9w)

[2.1.2. Cơ chế phân tán: 5](#_e4fmo8fgcgxp)

[2.1.3 Cơ chế bảo mật 5](#_481x2goihh4h)

[2.2 Mô tả cơ chế phân tích 5](#_1dmgpc6qzses)

[2.3 Cài đặt các cơ chế 7](#_s63gx9anah6d)

[2.3.1 Cơ chế bảo mật 7](#_k76dojjtxhua)

[2.3.1.1 Khung nhìn tĩnh 7](#_lfc4g25uvlfv)

[2.3.1.2 Khung nhìn động: Cài đặt bảo mật người dùng 9](#_lxdz0b1ijx5k)

[2.3.1.3 Khung nhìn động: Bảo mật việc truy nhập dữ liệu 10](#_id4jg9bdm82g)

[2.3.2. Lưu trữ – OODBMS - Lưu trữ các đối tượng 10](#_5y0qlyoawmr1)

[2.3.3. Cơ chế phân tán 10](#_bfgjwny3dic7)

[**3. Khung nhìn logic 10**](#_yfhhgkgj03ri)

[3.1 Phân tích kiến trúc 10](#_8hv0vzsen6m)

[3.1.1. Các trừu tượng chính 10](#_r4km4h64gmw8)

[3.1.2 Tổ chức bậc cao của mô hình và các phụ thuộc: 11](#_wsl1dn2mctpr)

[3.1.3 Các gói và các phụ thuộc 13](#_pvm3qnrprdnp)

[3.1.3.1 Các lớp nhận dạng 13](#_ee6djsleytoi)

[3.1.3.1 Xác định các packages 15](#_lt9dm99p5idw)

[3.1.3.3 Biểu đồ phụ thuộc 17](#_e4oy7fwand5b)

[3.1.3.4 Mô tả các gói: 17](#_x209rp1ryxg4)

[**4. Thiết kế Cơ sở dữ liệu. 18**](#_i0a4mf55o8g8)

[4.1 Xác định các đối tượng của bài toán và các mối quan hệ giữa chúng. 18](#_2okmih7h1tep)

[4.1.1. Các đối tượng và thuộc tính: 18](#_7bf330n3c06s)

[4.1.2 Các mối quan hệ giữa các đối tượng 19](#_4b2xbpe7maaj)

[4.2 Vẽ sơ đồ ERD (Entity Relationship Diagram) 21](#_jkehednxi0h0)

[4.3. Chuyển từ ERD sang ER 21](#_2mpsv7f3zw1l)

# **Giới thiệu**

## **1.1 Mục đích**

* Tài liệu này bổ sung tài liệu cho hệ thống đặt xe trực tuyến GRIB do nhóm 3 thực hiện trong bối cảnh môn học Phân tích và thiết kế hướng đối tượng.
* Tài liệu này được viết dựa trên định dạng báo cáo “IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998”.
* Dựa trên nội dung tài liệu đồng nhất và không có mâu thuẫn với các tài liệu khác của hệ thống GRIB. Nó cung cấp các thông tin kiến trúc hỗ trợ phát triển thiết kế Hệ thống GRIB. Điều này là do môn học OOAD tập trung vào việc chứng minh kiến trúc ảnh hưởng như thế nào đến mô hình thiết kế.

## **1.2 Quy ước tài liệu**

Không có

## **1.3 Độc giả và cách đọc tài liệu**

Các loại người đọc khác nhau mà tài liệu hướng tới là:

* Nhà thiết kế: có thiết kế phải đáp ứng các yêu cầu quy định tại SRS này.
* Developers: Dev là người triển khai hệ thống từ khâu thiết kế, tài liệu thành một phiên bản có thể chạy được. Dev phải đọc toàn bộ tài liệu để thực hiện đúng hệ thống.
* Người kiểm thử: người phải đảm bảo rằng các yêu cầu có thể xác thực được và các kịch bản kiểm thử của họ phải xác thực các yêu cầu. Người kiểm thử nên đọc chi tiết để viết kịch bản kiểm thử đơn vị cụ thể.
* Khách hàng: Khách hàng là người đặt mua hệ thống và muốn có một hệ thống mới tốt hơn hệ thống cũ. Trong khóa học này, khách hàng có thể được coi là một công ty cung cấp dịch vụ xe ôm.
* Người viết tài liệu: người sẽ viết tài liệu trong tương lai (báo cáo, biên bản).
* Người dùng: là bất xế người dùng hoặc các tài xế tham gia vào quá trình sử dụng GRIB.

## **1.4 Phạm vi dự án**

* Ban đầu, người dùng của phần mềm là các khách hàng và lái xe. Phần mềm sẽ tạo ra môi trường để khách hàng có thể đặt xe để di chuyển hoặc để ship hàng.
* Bên cạnh đó, phần mềm cũng hướng tới những người muốn kiếm thêm thu nhập bằng cách đăng kí làm tài xế trên hệ thống, nhận chở khách và ship hàng theo yêu cầu của khách hàng.

## **1.5 Tài liệu tham khảo**

* [1] Form of presentation IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.
* [2] CariGO Architecture Handbook, payroll\_arc\_handbk\_rpt.pdf
* [3] Glossary for GRIB, OOAD Nhóm 3
* [4] System Architectural Analysis for GRIB, OOAD Nhóm 3.
* [5] <http://sce.uhcl.edu/helm/rationalunifiedprocess/process/workflow/ana_desi/co_amech.htm>

# **Các cơ chế kiến trúc**

## **2.1 Các cơ chế phân tích**

### **2.1.1 Cơ chế lưu trữ**

| Lớp | Customer | Driver | Trip | Request |
| --- | --- | --- | --- | --- |
| Kích cỡ | 300KB/đối  tượng | 1500KB/đối  tượng | 100KB/chuyến  đi | 3KB-5KB/đối  tượng |
| Dung lượng | 1000000 đối  tượng | 50000 đối tượng | 3 triệu chuyến | 3 triệu đối tượng |
| Tần suất  truy cập | + Tạo mới: 1000  đối tượng/tuần  + Cập nhật:  10000 đối  tượng/tuần | + Tạo mới: 100 đối tượng/tuần  + Cập nhật:  200/tuần | + Tạo mới:  100000  chuyến/tuần  + Cập nhật:  2000 chuyến | + Tạo mới:  100000 đối  tượng/tuần  + Cập nhật:  2000 đối tượng |

### **2.1.2. Cơ chế phân tán:**

* Dữ liệu được lưu trữ trên Cơ sở dữ liệu MySQL, các thuộc tính thường xuyên được truy cập ví dụ như phiên đăng nhập, các địa điểm ưa thích của khách hàng có thể được lưu trữ trên Redis.

### **2.1.3 Cơ chế bảo mật**

* Data granularity: Cấp thuộc tính
* User granularity: 3 Đối tượng: Khách hàng, Nhân viên, Lái xe.
* Security rules:
* Chỉ khi đã đăng ký, khách hàng/nhân viên/lái xe mới có thể đăng nhập vào hệ thống.
* Chỉ khi đã đăng nhập vào hệ thống thì khách hàng có thể đặt xe cho chuyến đi.
* Chỉ khi đã đăng nhập vào hệ thống thì khách hàng có thể sửa thông tin cá nhân.
* Chỉ có duy nhất nhân viên có thể thêm bớt, xóa thông tin của lái xe trên hệ thống.

## **2.2 Mô tả cơ chế phân tích**

Ánh xạ lớp phân tích – cơ chế phân tích

| STT | Lớp phân tích | Cơ chế phân tích |
| --- | --- | --- |
| 1 | Customer | Persistency, Security |
| 2 | Driver |
| 3 | Staff |
| 4 | Request |
| 5 | Trip |
| 6 | DriverWallet |
| 7 | Wallet |
| 8 | Notification |
| 9 | TransactionBilling |
| 10 | TripBilling |
| 11 | Vehicles |
| 12 | BillingSystem | Persistency, Legacy Interface, Security |
| 13 | MapAPI | Distribution, Legacy Interface |
| 14 | LoginController | Distribution |
| 15 | BookingController |
| 16 | RouteTrackingController |
| 17 | FeedbackController |
| 18 | ManageWalletController |
| 19 | AcceptRequestController |
| 20 | PaymentController |
| 21 | ViewNotificationController |
| 22 | ViewTripHistoryController |
| 23 | ManageDriverInfoController |
| 24 | ManageCustomerInfoController |
| 25 | CreateNotificationController |
| 26 | LoginForm | Error detection /handling /reporting |
| 27 | CustomerLoginForm |
| 28 | BookingForm |
| 29 | CustomerRouteTrackingForm |
| 30 | DriverRouteTrackingForm |
| 31 | FeedbackForm |
| 32 | ManageWalletForm |
| 33 | WalletForm |
| 34 | AcceptRequestForm |
| 35 | CustomerPaymentForm |
| 36 | DriverPaymentForm |
| 37 | ViewNotificationForm |
| 38 | ViewTripHistoryForm |
| 39 | ManageCustomerInfoForm |
| 40 | ManageDriverInfoForm |
| 41 | CreateNotificationForm |

Ánh xạ cơ chế phân tích - thiết kế - triển khai

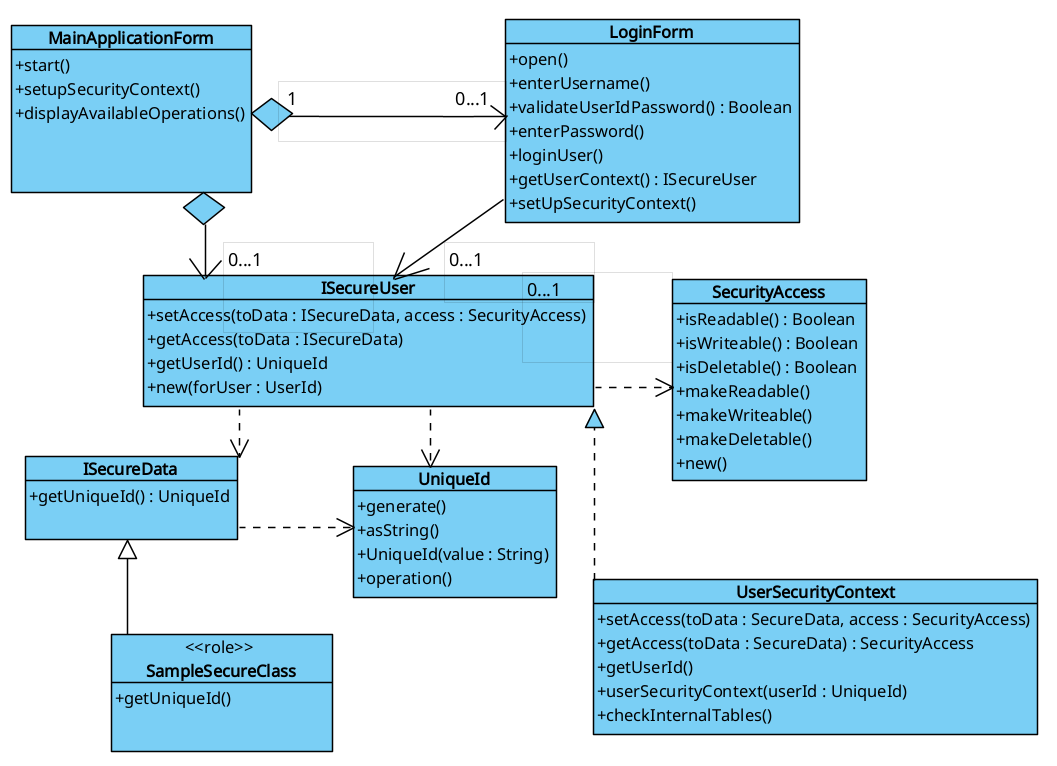
| **Cơ chế phân tích** | **Cơ chế thiết kế** | **Cơ chế cài đặt/ triển khai** |
| --- | --- | --- |
| Persistency | Relational Database Management System | Spring Data JPA |
| Security | Json Web tokens | Java Spring framework |
| Distribution | Remote Method Invocation  (RMI) | Java Spring framework |

## **2.3 Cài đặt các cơ chế**

### **2.3.1 Cơ chế bảo mật**

#### **2.3.1.1 Khung nhìn tĩnh**

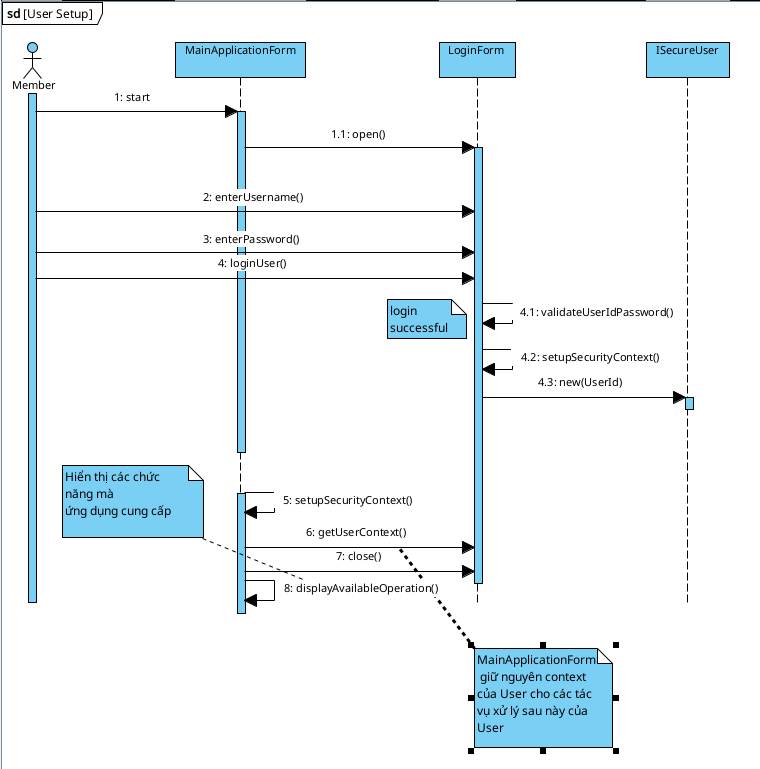
2.3.1.1.1. Biểu đồ lớp



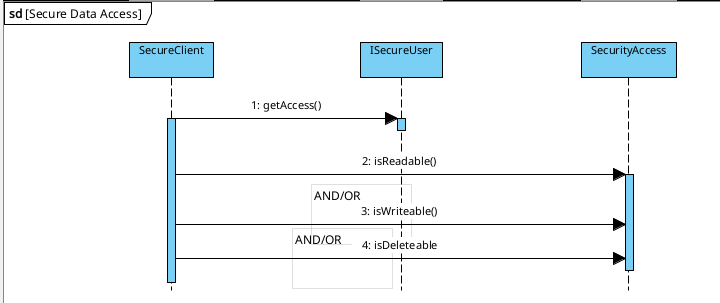
2.3.1.1.2. Mô tả các lớp

* ISecureData: Cơ chế phân tích: Bảo mật
* SecurityAccess: Cơ chế phân tích: Bảo mật
* UserSecurityContext: Cơ chế phân tích: Bảo mật
* UniqueId: Cơ chế phân tích: Bảo mật
* MainApplicationForm: Truy vết yêu cầu: Tính khả dụng
* ISecureUser: Cơ chế phân tích: Bảo mật
* LoginForm: Cơ chế phân tích: Bảo mật

#### **2.3.1.2 Khung nhìn động:** Cài đặt bảo mật người dùng



#### **2.3.1.3 Khung nhìn động: Bảo mật việc truy nhập dữ liệu**



### **2.3.2. Lưu trữ – OODBMS - Lưu trữ các đối tượng**

(đang trong quá trình cập nhật…)

### **2.3.3. Cơ chế phân tán**

(đang trong quá trình cập nhật…)

# **3**. **Khung nhìn logic**

## **3.1 Phân tích kiến trúc**

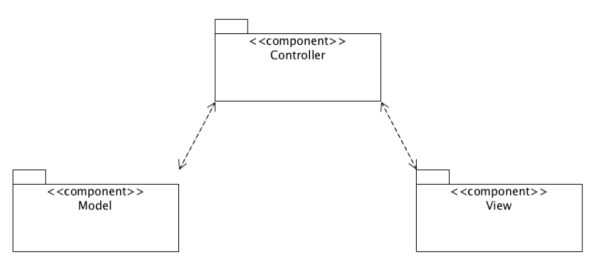
### **3.1.1. Các trừu tượng chính**

| STT | Trừu tượng chính | Định nghĩa |
| --- | --- | --- |
| 1 | Customer | Tài khoản cho khách hàng, mỗi khách hàng có ID duy nhất, số điện thoại duy nhất dùng để đăng  nhập vào hệ thống. |
| 2 | Driver | Tài khoản cho tài xế, mỗi tài xế có ID duy nhất, có tên và mật khẩu dùng để đăng nhập vào hệ thống |
| 3 | Staff | Tài khoản cho nhân viên, mỗi nhân viên có ID duy nhất, có tên và mật khẩu dùng để đăng nhập vào hệ thống |
| 4 | Request | Yêu cầu về việc đi lại được tạo ra bởi khách hàng và được chấp nhận và thực hiện bởi tài xế |
| 5 | Trip | Lưu lại thông tin về chuyến đi của tài xế và khách hàng |
| 6 | PayWallet | Ví điện tử của khách hàng, là duy nhất với mỗi  khách hàng |
| 7 | DriverWallet | Ví điện tử của tài xế, là duy nhất với mỗi tài xế. |
| 8 | Vehicles | Phương tiện vận chuyển của tài xế. |
| 9 | TripBilling | Thông tin thanh toán cho mỗi chuyến đi. |
| 10 | TransactionBilling | Thông tin chuyển tiền, rút tiền,… của ví điện tử. |
| 11 | Notification | Thông báo cho khách hàng và tài xế. |

### **3.1.2 Tổ chức bậc cao của mô hình và các phụ thuộc:**

MVC là mô hình thiết kế được sử dụng phổ biến trong các ứng dụng hiện

nay. MVC tổ chức các sắp xếp mã nguồn một cách gọn gàng, điều này giúp cho lập trình viên nhìn vào ứng dụng và chia sẻ ứng dụng với người khác dễ dàng và gọn hơn nhièu. Mô hình này tách ứng dụng thàng 3 phần riêng biệt với nhau, mỗi thành phần thực hiện nhiệm vụ riêng biệt và độc lập với các thành phần khác. MVC bao gồm Model – mô hình (dữ liệu), View – chế độ xem (giao diện người dùng) và Controller – bộ điều khiển (các quy trình xử lý đầu vào). Các thành phần và các thức hoạt động mô hình MVC được mô tả dưới đây:



Các thành phần:

* Model: Là thành phần chịu trách nhiệm quản lý dữ liệu, nó lưu trữ dữ

liệu và các tính toán xử lý logic để xử lý logic truy xuất các thực thể từ cơ sở dữ

liệu. Có những cơ sở dữ liệu phổ biến như MySQL, SQL Server, PostgreSQL,...

* View: Là thành phần chịu trách nhiệm hiển thị dữ liệu đã được truy xuất

từ model dưới một định dạng nào đó như form, textbox, … Còn về cách sử dụng

của View thì View thực hiện format lại dữ liệu được gửi từ controller và trình bày

theo 1 định dạng (file html).

* Controller: Là thành phần trung gian, thực hiện nhiệm vụ xử lý cho

model và view tương tác với nhau. Controller nhận các dữ liệu từ người dùng đưa

vào thông qua các phương tiện chuột, bàn phím, hoặc các tương tác thực hiện trên

control, … sau đó gọi tới các model để thực hiện các yêu cầu xử lý dữ liệu và gửi

các dữ liệu đã được truy xuất ra ngoài View.

Dưới đây là mô tả cách mô hình MVC làm việc:

* Người dùng tương tác với màn hình View thông qua việc click vào button, người dùng sẽ gửi yêu cầu đi.
* Controller nhận yêu cầu và điều hướng đến đúng phương thức xử lý ở Model.
* Model nhận thông tin và thực thi xử lý dữ liệu các yêu cầu.
* Khi Model hoàn tất việc xử lý, View sẽ nhận kết quả từ Model và hiển thị lại cho người dùng.

### **3.1.3 Các gói và các phụ thuộc**

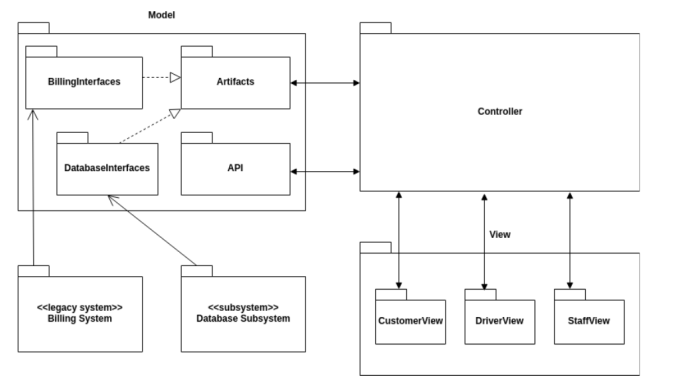
#### **3.1.3.1 Các lớp nhận dạng**

| STT | Lớp phân tích | Phần tử thiết kế |
| --- | --- | --- |
| 1 | Customer | Customer, Database Subsystem |
| 2 | Driver | Driver, Database Subsystem |
| 3 | Staff | Staff, Database Subsystem |
| 4 | Request | Request, Database Subsystem |
| 5 | Trip | Trip, Database Subsystem |
| 6 | DriverWallet | DriverWallet, Database Subsystem |
| 7 | Wallet | Wallet, Database Subsystem |
| 8 | Notification | Notification, Database Subsystem |
| 9 | Vehicles | Vehicles, Database System |
| 10 | TransactionBilling | TransactionBilling, Billing System |
| 11 | TripBilling | TripBilling, Billing System |
| 12 | BillingSystem | IBillingSystem |
| 13 | MapAPI | MapAPI |
| 14 | LoginController | Ánh xạ trực tiếp tới các lớp thiết kế |
| 15 | BookingController |
| 16 | RouteTrackingController |
| 17 | FeedbackController |
| 18 | ManageWalletController |
| 19 | AcceptRequestController |
| 20 | ManageWalletController |
| 21 | PaymentController |
| 22 | ViewNotificationController |
| 23 | ViewTripHistoryController |
| 24 | ManageDriverInfoController |
| 25 | ManageCustomerInfoController |
| 26 | CreateNotificationController |
| 27 | LoginForm |
| 28 | CustomerLoginForm |
| 29 | BookingForm |
| 30 | CustomerRouteTrackingForm |
| 31 | DriverRouteTrackingForm |
| 32 | FeedbackForm |
| 33 | ManageWalletForm |
| 34 | WalletForm |
| 35 | AcceptRequestForm |
| 36 | ManageWalletForm |
| 37 | CustomerPaymentForm |
| 38 | DriverPaymentForm |
| 39 | ViewNotificationForm |
| 40 | ViewTripHistoryForm |
| 41 | ManageCustomerInfoForm |
| 42 | ManageDriverInfoForm |
| 43 | CreateNotificationForm |

#### **3.1.3.1 Xác định các packages**

| STT | Phần tử thiết kế | Package |
| --- | --- | --- |
| 1 | Customer | Model::Artifacts |
| 2 | Driver | Model::Artifacts |
| 3 | Staff | Model::Artifacts |
| 4 | Request | Model::Artifacts |
| 5 | Trip | Model::Artifacts |
| 6 | DriverWallet | Model::Artifacts |
| 7 | Wallet | Model::Artifacts |
| 8 | Notification | Model::Artifacts |
| 9 | Vehicles | Model::Artifacts |
| 10 | TransactionBilling | Model::Artifacts |
| 11 | TripBilling | Model::Artifacts |
| 12 | BillingSystem | BillingSystem |
| 13 | MapAPI | Model::API |
| 14 | LoginController | Controller |
| 15 | BookingController | Controller |
| 16 | RouteTrackingController | Controller |
| 17 | FeedbackController | Controller |
| 18 | ManageWalletController | Controller |
| 19 | AcceptRequestController | Controller |
| 20 | ManageDriverWalletController | Controller |
| 21 | PaymentController | Controller |
| 22 | ViewNotificationController | Controller |
| 23 | ViewTripHistoryController | Controller |
| 24 | ManageDriverInfoController | Controller |
| 25 | ManageCustomerInfoController | Controller |
| 26 | CreateNotificationController | Controller |
| 27 | LoginForm | View::Staff |
| 28 | CustomerLoginForm | View::Customer |
| 29 | BookingForm | View::Customer |
| 30 | CustomerRouteTrackingForm | View::Customer |
| 31 | DriverRouteTrackingForm | View::Driver |
| 32 | FeedbackForm | View::Customer |
| 33 | ManageWalletForm | View::Customer |
| 34 | WalletForm | View::Driver |
| 35 | AcceptRequestForm | View::Driver |
| 36 | ManageDriverWalletForm | View::Driver |
| 37 | CustomerPaymentForm | View::Customer |
| 38 | DriverPaymentForm | View::Driver |
| 39 | ViewNotificationForm | View::Customer |
| 40 | ViewTripHistoryForm | View::Customer |
| 41 | ManageCustomerInfoForm | View::Staff |
| 42 | ManageDriverInfoForm | View::Staff |
| 43 | CreateNotificationForm | View::Staff |
| 44 | ICustomerDatabase | Model::DatabaseInterfaces |
| 45 | IDriverDatabase | Model::DatabaseInterfaces |
| 46 | IStaffDatabase | Model::DatabaseInterfaces |
| 47 | IRequestDatabase | Model::DatabaseInterfaces |
| 48 | ITripDatabase | Model::DatabaseInterfaces |
| 49 | IDriverWalletDatabase | Model::DatabaseInterfaces |
| 50 | IWalletDatabase | Model::DatabaseInterfaces |
| 51 | INotificationDatabase | Model::DatabaseInterfaces |
| 52 | IVehiclesDatabase | Model::DatabaseInterfaces |
| 53 | ITransactionBilling | Model::BillingInterfaces |
| 54 | ITripBilling | Model::BillingInterfaces |

#### **3.1.3.3 Biểu đồ phụ thuộc**



#### **3.1.3.4 Mô tả các gói:**

- Model: Gói này có thể đại diện cho mô hình dữ liệu cho hệ thống. Trong hệ thống đặt xe FRIB, gói có thể bao gồm thông tin về người dùng (người lái và tài xế), phương tiện, giá vé và chuyến đi.

- BillingInterfaces: Gói này có thể xác định các giao diện để tương tác với hệ thống thanh toán. Trong hệ thống đặt xe GRIB, gói này có thể bao gồm các chức năng tính giá vé, xử lý thanh toán và xử lý hoàn tiền.

- Tạo phẩm: Gói này có thể bao gồm những thứ như tệp lịch sử các chuyến xe, tệp cấu hình hoặc tệp lược đồ cơ sở dữ liệu.

- Controller: Gói này có thể đại diện cho một phần của hệ thống xử lý sự tương tác của người dùng và kiểm soát hệ thống. Trong hệ thống này, gói này có thể bao gồm các chức năng xử lý các yêu cầu của người dùng về đặt chuyến đi, hủy chuyến đi , tìm đường đi, quản lý tài khoản…

- Database Interfaces: Gói này có thể xác định các giao diện để tương tác với cơ sở dữ liệu. Trong hệ thống này, gói này có thể bao gồm các chức năng thêm, cập nhật và xóa dữ liệu về người dùng, phương tiện, chuyến đi và các thực thể khác.

- API: Gói này có thể đề cập đến Giao diện lập trình ứng dụng, cho phép các hệ thống khác tương tác với hệ thống đặt xe. Ví dụ: dịch vụ bản đồ có thể sử dụng GoogleMap API để cung cấp chỉ đường cho người lái xe.

- BillingSystem: Đảm nhiệm cho tác vụ thanh toán cước phí của hệ thống, đảm bảo các giao dịch được chính xác và an toàn, có thể kết hợp với các ví điện tử như Momo, VNPay,...

- Database subsystem: Hệ thống con lưu trữ toàn bộ dữ liệu của hệ thống, sử dụng MySQL.

- Views: Các gói con bên trong bao gồm CustomerView, DriverView, StaffView chứa các thành phần giao diện tương ứng với các vai trò khác nhau trong hệ thống.

# 4. **Thiết kế Cơ sở dữ liệu.**

## **4.1 Xác định các đối tượng của bài toán và các mối quan hệ giữa chúng.**

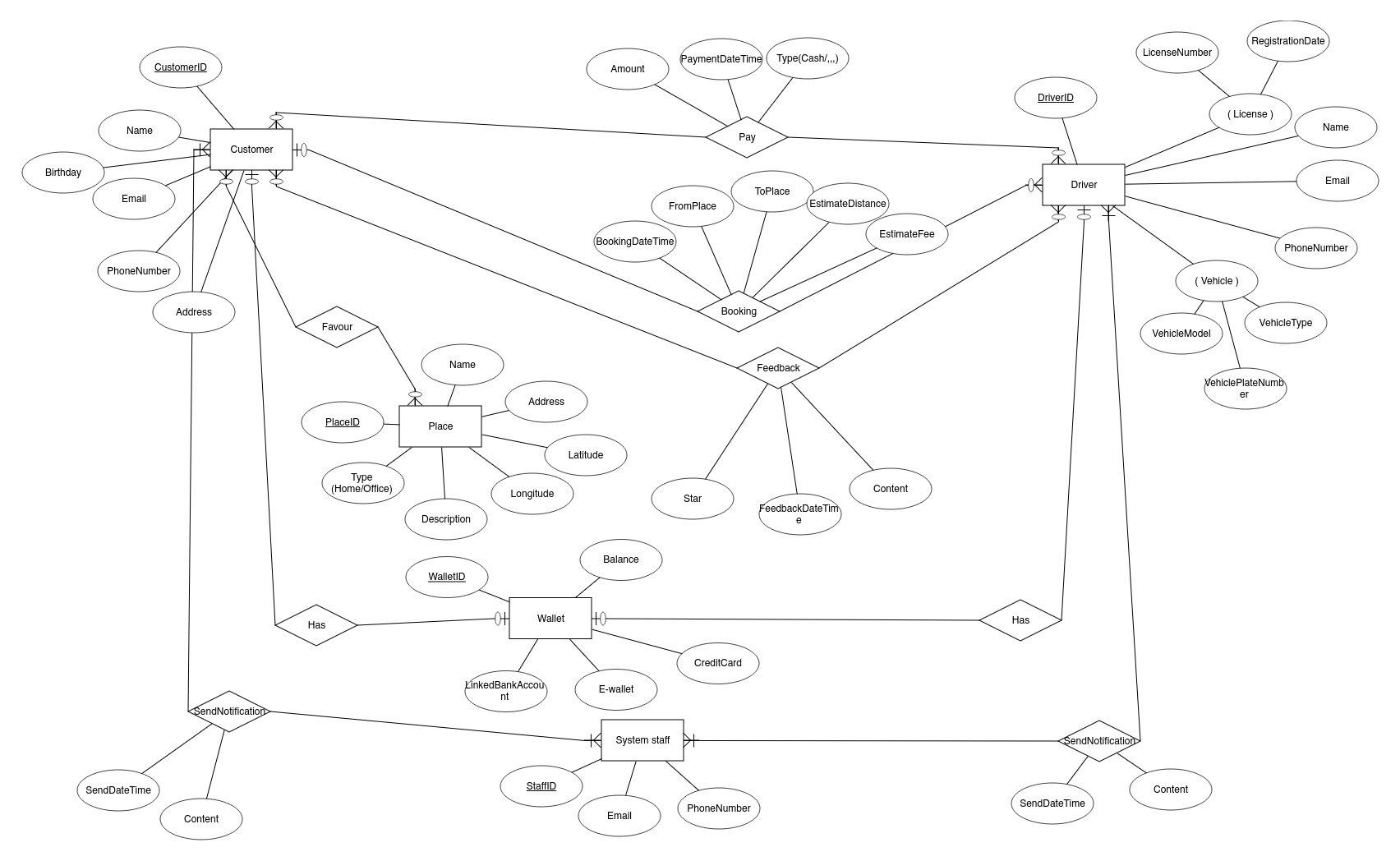
#### **4.1.1. Các đối tượng và thuộc tính:**

* **Khách hàng:**
  + Mã số khách hàng
  + Tên khách hàng
  + Ngày sinh
  + Email
  + Địa chỉ
  + Số điện thoại
* **Tài xế:**
  + Mã số tài xế
  + Tên tài xế
  + Email
  + Số điện thoại
  + Phương tiện đăng kí (Ô tô hay xe máy, Đời bao nhiêu, Biển số xe)
  + Bằng lái (Mã số bằng lái, ngày tháng cấp bằng)
* **Ví điện tử:**
  + Mã số ví
  + Số dư
  + Ngân hàng liên kết
  + Thẻ tín dụng liên kết
* **Các địa điểm**
  + Tên địa điểm
  + Địa chỉ
  + Kinh độ
  + Vĩ độ
  + Mô tả địa điểm
  + Loại địa điểm (Nhà riêng/ Văn phòng)
* **Nhân viên hệ thống**
  + Tên
  + Email

#### **4.1.2 Các mối quan hệ giữa các đối tượng**

* **Khách hàng (Đặt) Tài xế:**
  + Tại một thời điểm, 1 tài xế có thể nhận được nhiều yêu cầu đặt xe từ khách hàng, nhưng một khách hàng chỉ có thể đặt được một tài xế vào thời điểm đó. Tuy nhiên, trong quá trình sử dụng hệ thống, một khách hàng có thể đặt nhiều tài xế khác nhau (miễn là không cùng một thời điểm) và một tài xế có thể nhận đặt xe từ nhiều khách hàng khác nhau => **Mối quan hệ N:N.**
  + **Các thuộc tính của quan hệ:** 
    - Ngày giờ đặt xe
    - Đi từ đâu
    - Đến đâu
    - Khoảng cách ước tính
    - Chi phí ước tính.
* **Khách hàng (Đánh giá) Tài xế:** 
  + Một khách hàng có thể đánh giá nhiều tài xế, một tài xế có thể đánh giá được nhiều khách hàng, miễn là khách hàng đó đã đặt xe của tài xế đó. => **Mối quan hệ N:N.**
  + **Các thuộc tính của quan hệ:**
    - Số sao mà khách hàng đánh giá (4 sao, …).
    - Nội dung phản ánh.
    - Ngày giờ phản ánh.
* **Khách hàng (Thanh toán cho) Tài xế:** 
  + Một khách hàng có thể thanh toán cho nhiều tài xế, một tài xế có thể nhận thanh toán từ nhiều khách hàng, miễn là khách hàng đó đã đặt xe của tài xế đó => **Mối quan hệ N:N.**
  + **Các thuộc tính của quan hệ:** 
    - Số tiền khách hàng thanh toán
    - Thời gian thanh toán
    - Hình thức thanh toán (Tiền mặt, chuyển khoản)
* **Khách hàng (Ưa thích) Địa điểm:** 
  + Một khách hàng có thể yêu thích nhiều địa điểm, một địa điểm có thể có nhiều khách hàng ưa thích. => **Mối quan hệ N:N.**
* **Khách hàng (Sở hữu) Ví:** Một khách hàng chỉ sở hữu một ví trên hệ thống, một ví trên hệ thống cũng chỉ được sở hữu bởi 1 khách hàng duy nhất => **Mối quan hệ 1:1.**
* **Tài xế (Sở hữu) Ví:** Một tài xế chỉ sở hữu một ví trên hệ thống, một ví trên hệ thống cũng chỉ được sở hữu bởi 1 tài xế duy nhất => **Mối quan hệ 1:1.**
* **Nhân Viên Hệ Thống (Gửi thông báo cho) Khách hàng:** 
  + Nhân viên hệ thống có thể gửi thông báo tới nhiều khách hàng, 1 khách hàng có thể nhận thông báo từ nhiều nhân viên hệ thống. => **Mối quan hệ N:N.**
  + **Các thuộc tính của quan hệ:**
    - Ngày giờ gửi thông báo.
    - Nội dung thông báo
* **Nhân Viên Hệ Thống (Gửi thông báo cho) Tài xế:** 
  + - Nhân viên hệ thống có thể gửi thông báo tới nhiều tài xế, tài xế có thể nhận thông báo từ nhiều nhân viên hệ thống => Mối quan hệ **N:N.**
  + **Các thuộc tính của quan hệ:**
    - Ngày giờ gửi thông báo.
    - Nội dung thông báo

## **4.2 Vẽ sơ đồ ERD (Entity Relationship Diagram)**

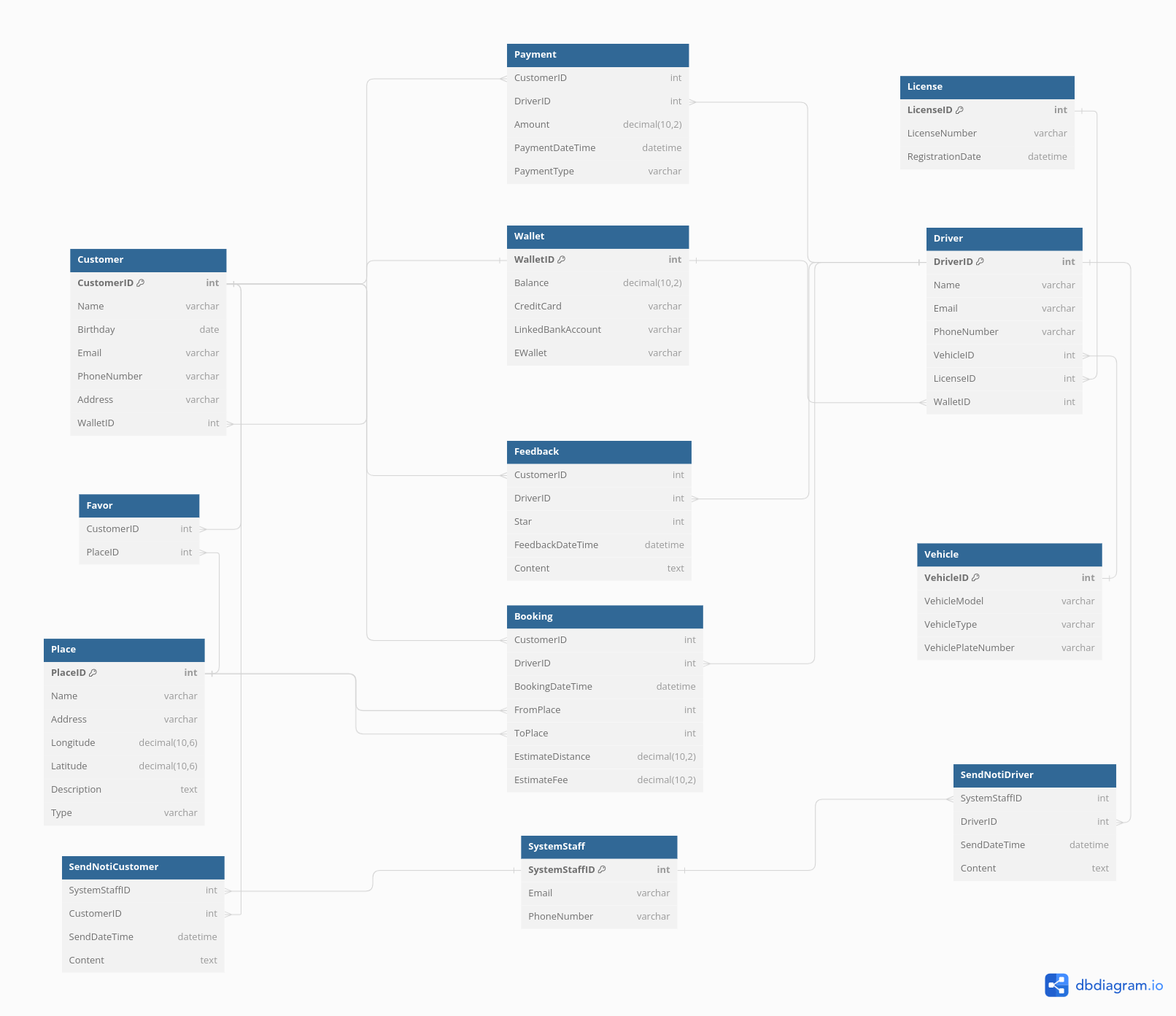
Từ việc xác định các thực thể và các mối quan hệ, ta có sơ đồ ERD: 

## **4.3. Chuyển từ ERD sang ER**

* **Chuyển đổi các Thực thể mạnh sang các Quan hệ:**
  + **Customer (CustomerID, Name, Birthday, Email, PhoneNumber, Address).**
  + **Driver (DriverID, Name, Email, PhoneNumber)**, do **Vehicle** là thuộc tính phức hợp nằm trong **Driver**, nên ta tạo ra quan hệ **Vehicle (VehicleID, VehicleModel, VehicleType, VehiclePlateNumber)**. Sau đó đưa **VehicleID** vào trong quan hệ Driver. Tương tự đối với **License (LicenseID, LicenseNumber, RegistrationDate).** Khi này quan hệ Driver là: **Driver (DriverID, Name, Email, PhoneNumber, VehicleID, LicenseID).**
  + **SystemStaff (SystemStaffID, Email, PhoneNumber)**
  + **Place (PlaceID, Name, Address, Longtitude, Latitude, Description, Type).**
  + **Wallet (WalletID, Balance, CreditCard, LinkedBankAccount, E-wallet).**
* **Chuyển đổi các Mối quan hệ sang các Quan hệ:**
  + Quan hệ **Payment (CustomerID, DriverID, Amount, PaymentDateTime, PaymentType).**
  + Do mối quan hệ Đặt xe giữa Khách hàng và Tài xế là mối quan hệ N:N. Vì thế, ta sẽ tạo quan hệ **Booking (CustomerID, DriverID, BookingDateTime, FromPlace, ToPlace, EstimateDistance, EstimateFee).**
  + Quan hệ **Feedback** **(CustomerID, DriverID, Star, FeedbackDateTime, Content).**
  + Quan hệ **Favor (CustomerID, PlaceID).**
  + Mối quan hệ **Khách hàng sở hữu Ví** là mối quan hệ 1:1, cả Khách hàng và Ví đều tham gia toàn bộ, do đó ta có thể đưa khóa chính của Customer vào trong quan hệ Wallet hoặc ngược lại. Ở đây, ta đưa khóa chính của Wallet vào trong Customer => **Customer (CustomerID, Name, Birthday, Email, PhoneNumber, Address, WalletID).**
  + Tương tự như thế **Driver (DriverID, Name, Email, PhoneNumber, VehicleID, LicenseID, WalletID).**
  + **SendNotiCustomer (SystemStaffID, CustomerID, SendDateTime, Content).**
  + **SendNotiDriver (SystemStaffID, DriverID, SendDateTime, Content).**

**=> Ta sẽ có các bảng sau:**

* **Wallet (WalletID, Balance, CreditCard, LinkedBankAccount, E-wallet).**
* **Vehicle (VehicleID, VehicleModel, VehicleType, VehiclePlateNumber)**
* **License (LicenseID, LicenseNumber, RegistrationDate)**
* **Customer (CustomerID, Name, Birthday, Email, PhoneNumber, Address, WalletID)**
* **Driver (DriverID, Name, Email, PhoneNumber, VehicleID, LicenseID, WalletID).**
* **Place (PlaceID, Name, Address, Longtitude, Latitude, Description, Type).**
* **SystemStaff (SystemStaffID, Email, PhoneNumber)**
* **Payment (CustomerID, DriverID, Amount, PaymentDateTime, PaymentType).**
* **Booking (CustomerID, DriverID, BookingDateTime, FromPlace, ToPlace, EstimateDistance, EstimateFee).**
* **Feedback (CustomerID, DriverID, Star, FeedbackDateTime, Content).**
* **Favor (CustomerID, PlaceID).**
* **SendNotiCustomer (SystemStaffID, CustomerID, SendDateTime, Content).**
* **SendNotiDriver (SystemStaffID, DriverID, SendDateTime, Content).**

**Trực quan hóa thiết kế Cơ sở dữ liệu (MySQL)**