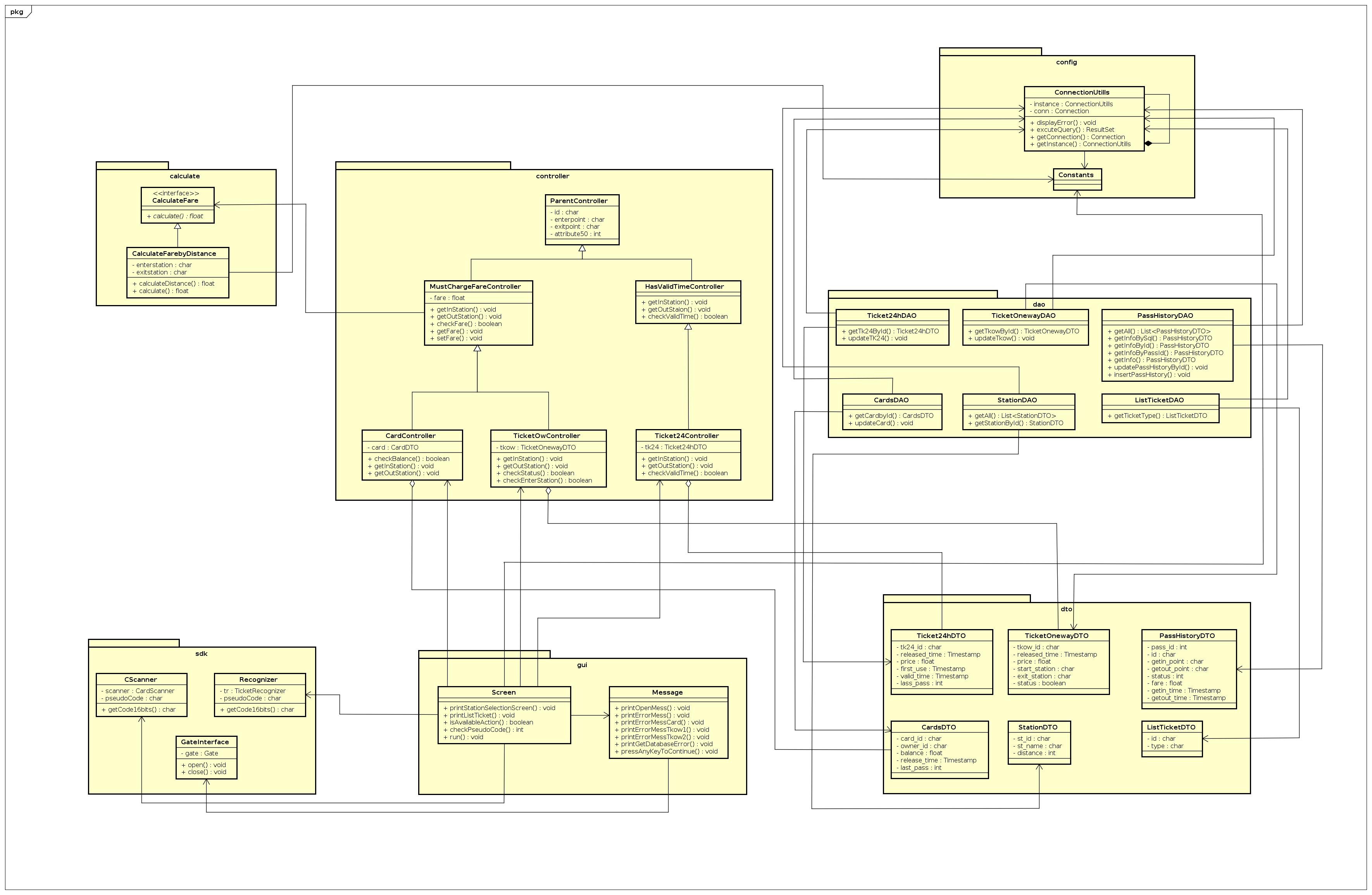
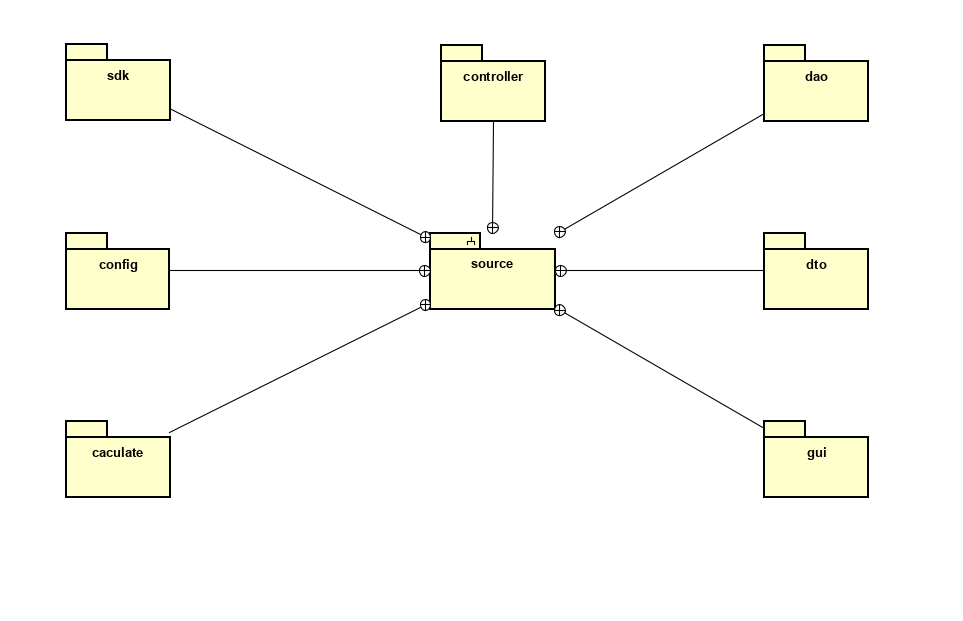
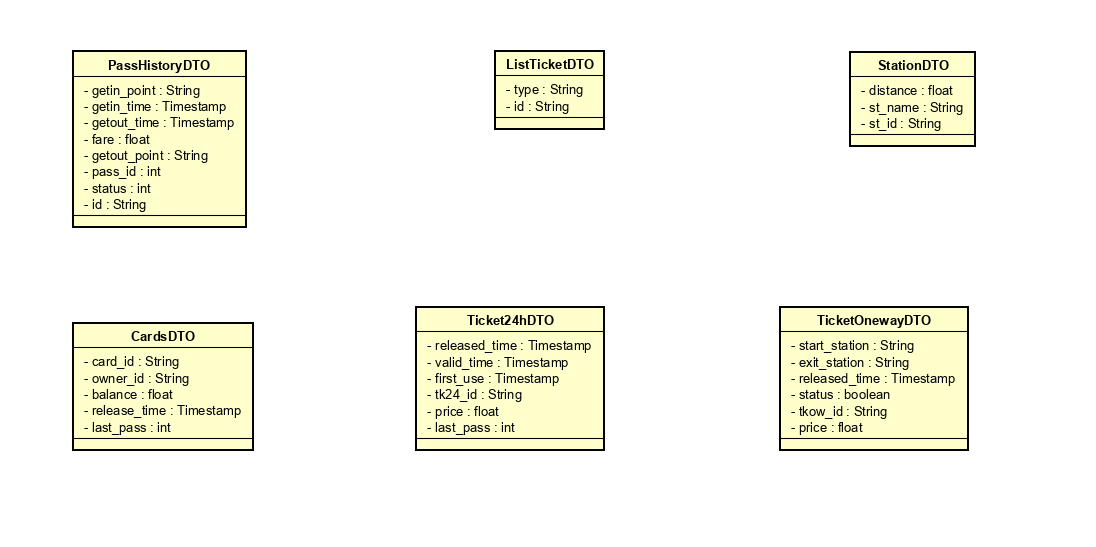
# Class diagram



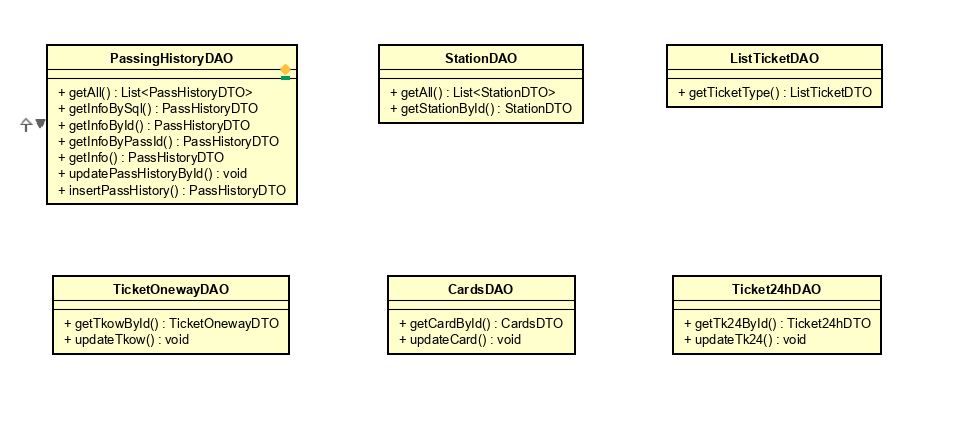
## Package diagram



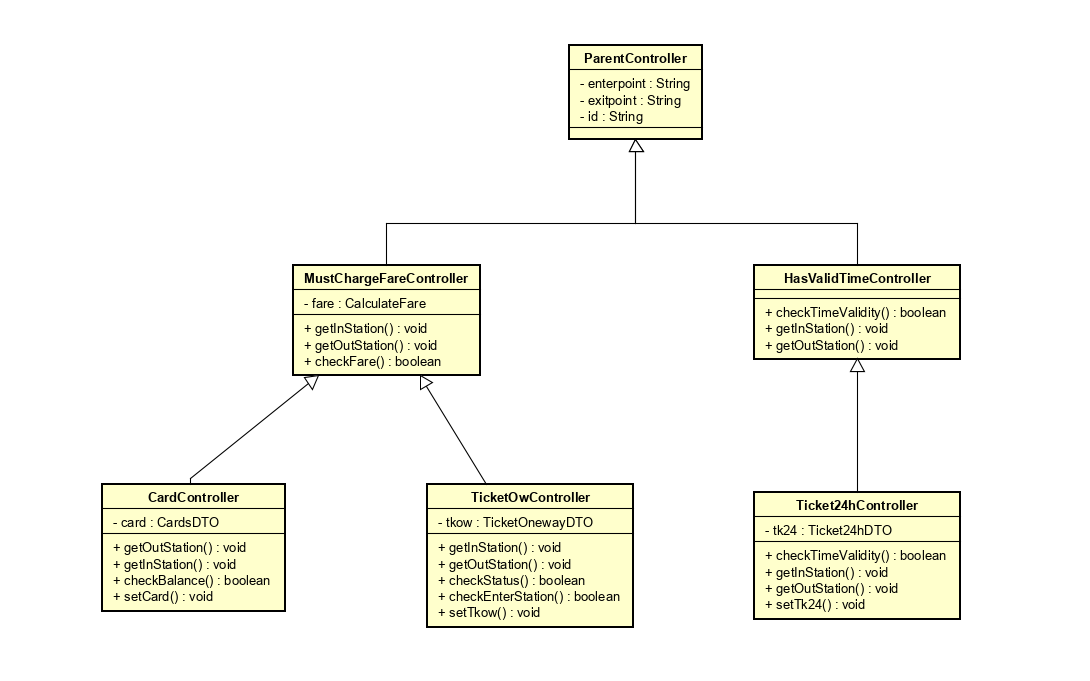
## Class diagram for package “dto”



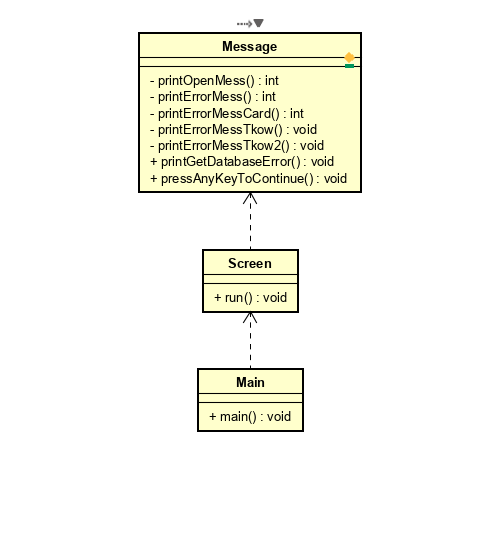
## Class diagram for package “dao”



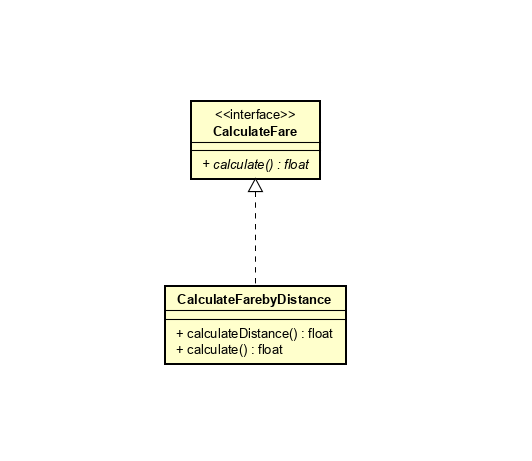
## Class diagram for package “controller”



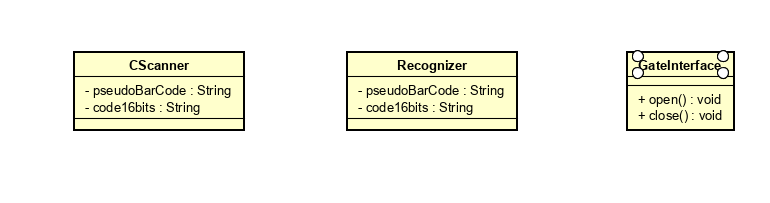
## Class diagram for package “gui”



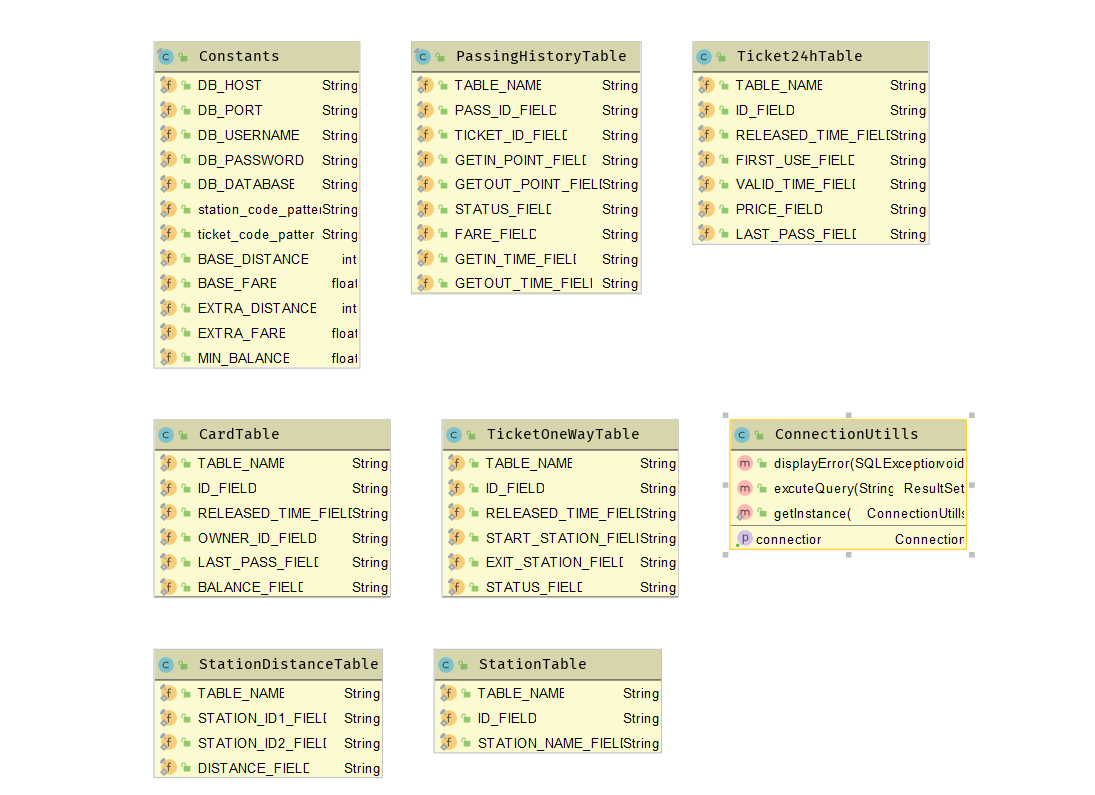
## Class diagram for package “calculate”



## Class diagram for package “sdk”



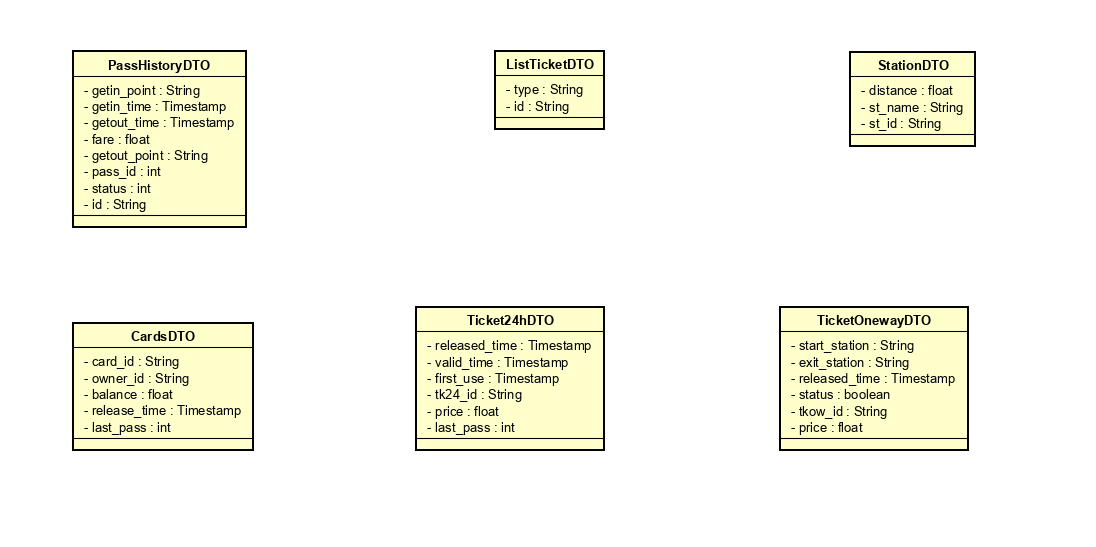
## Class diagram for package “config”



# Class design

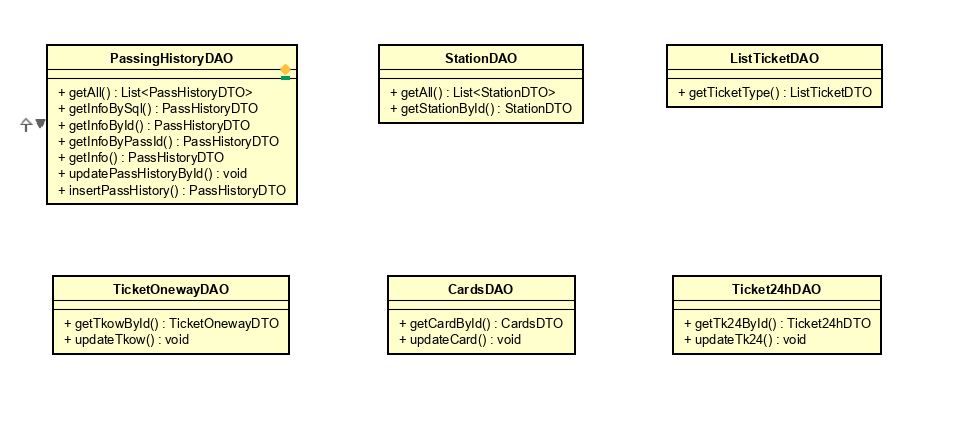
## Package “dto”

Mô tả: Là package chứa các class đại diện cho các loại loại vé, nhà ga, lịch sử ra vào

**Class**

|  |  |  |
| --- | --- | --- |
| *#* | *Name* | *Description* |
| 1 | PassHistoryDTO | Lưu thông tin một lần ra hoặc vào của hành khách |
| 2 | ListTicketDTO | Lưu ID và chủng loại của một vé hoặc một thẻ |
| 3 | StationDTO | Lưu thông tin của một nhà ga |
| 4 | CardsDTO | Lưu thông tin của một thẻ |
| 5 | Ticket24hDTO | Lưu thông tin của một vé 24h |
| 6 | TicketOnewayDTO | Lưu thông tin của một vé một chiều |

## Package “dao”

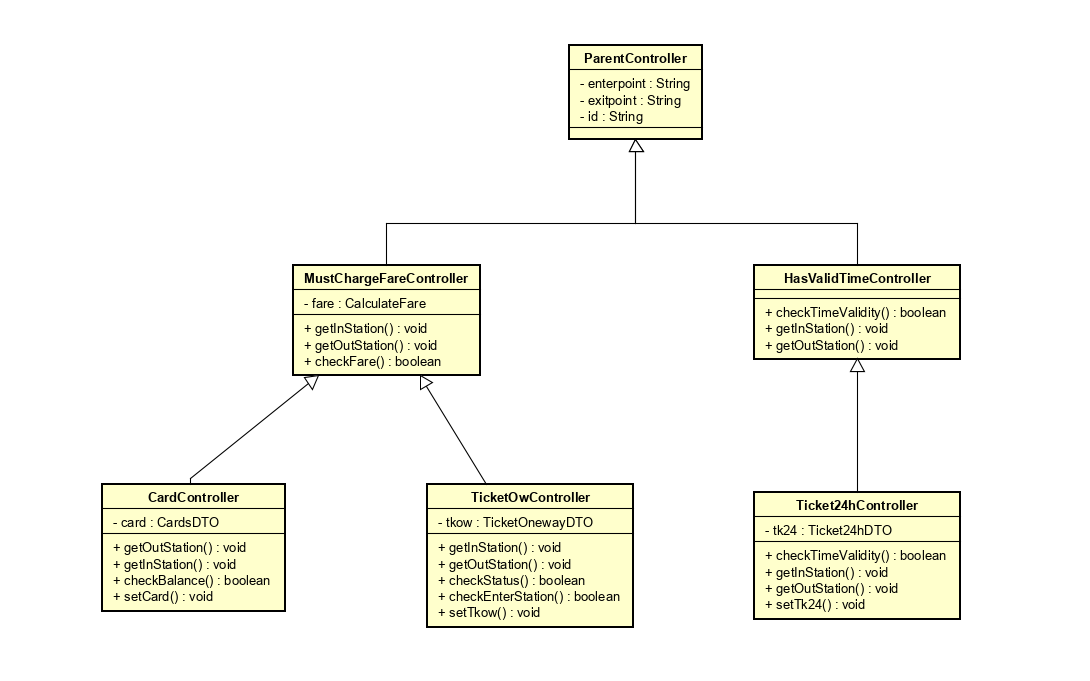


**Class**

|  |  |  |
| --- | --- | --- |
| *#* | *Name* | *Description* |
| 1 | PassHistoryDAO | Chưa các phương thức để lấy dữ liệu về lịch sử một lần ra hoặc vào của hành khách từ trong database ra và lưu vào một đối tượng PassHistoryDTO |
| 2 | ListTicketDAO | Chưa các phương thức để lấy dữ liệu về loại vé từ database và lưu vào 1 một đối tượng ListTicketDTO |
| 3 | StationDAO | Chưa các phương thức để lấy dữ liệu của một nhà ga từ database và lưu vào đối tượng Station DTO |
| 4 | CardsDAO | Chứa các phương thức để lấy dữ liệu của 1 thẻ từ database và lưu vào một đối tượng CardsDTO |
| 5 | Ticket24hDAO | Chứa các phương thức để lấy dữ liệu của 1 vé 24h từ database và lưu vào một đối tượng Ticket24hDTO |
| 6 | TicketOnewayDAO | Chứa các phương thức để lấy dữ liệu của 1 vé 1 chiều từ database và lưu vào một đối tượng TicketOnewayDTO |

## Package “controller”

Mô tả: chứa các class có nhiệm vụ thực hiện những thao tác nghiệp vụ khi có một hành khách đi qua máy soát vé

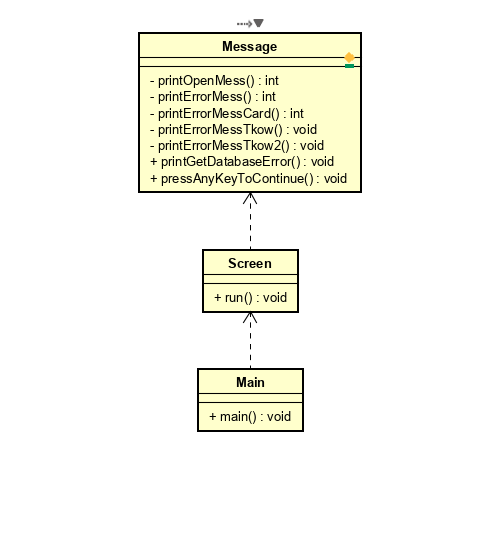


**Class**

|  |  |  |
| --- | --- | --- |
| *#* | *Name* | *Description* |
| 1 | ParrentController | Lớp trừu tượng đại diện cho một controller tổng quát |
| 2 | MustChargeFareController | Lớp trừu tượng đại diện cho controller chứa các phương thức để xử lý nghiệp vụ liên quan đến loại vé mà phải nạp tiền để trả phí cho mỗi lần đi |
| 3 | HasValidTimeController | Lớp trừu tượng đại diện cho controller chứa các phương thức để xử lý nghiệp vụ liên quan đến loại vé có thời hạn sử dụng |
| 4 | CardController | Chứa các phương thức để xử lý nghiệp vụ cho thẻ trả trước |
| 5 | Ticket24hController | Chứa các phương thức để xử lý nghiệp vụ cho vé 24h |
| 6 | TicketOwController | Chứa các phương thức để xử lý nghiệp vụ cho vé một chiều |

## Package “gui”

Mô tả: chứa các class có nhiệm vụ xử lý hiển thị giao diện người dùng



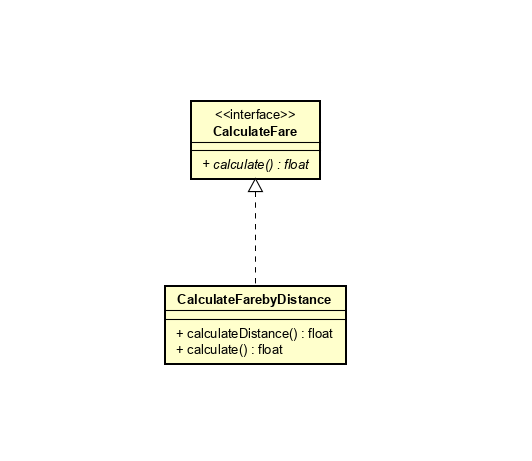
## Package “gui”

**Class**

|  |  |  |
| --- | --- | --- |
| *#* | *Name* | *Description* |
| 1 | Message | Chứa các phương thức để in ra màn hình thông báo thành công hoặc lỗi mỗi khi khách đi qua |
| 2 | Screen | Chứa các hàm hiển thị danh sách nhà ga, danh sách vé và xử lý dữ liệu nhập vào từ người dùng |
| 3 | Main | Chứa hàm main để khởi động chương trình |

## Package “calculate”

Mô tả: chứa các class đại diện cho các phương thức tính phí đi lại

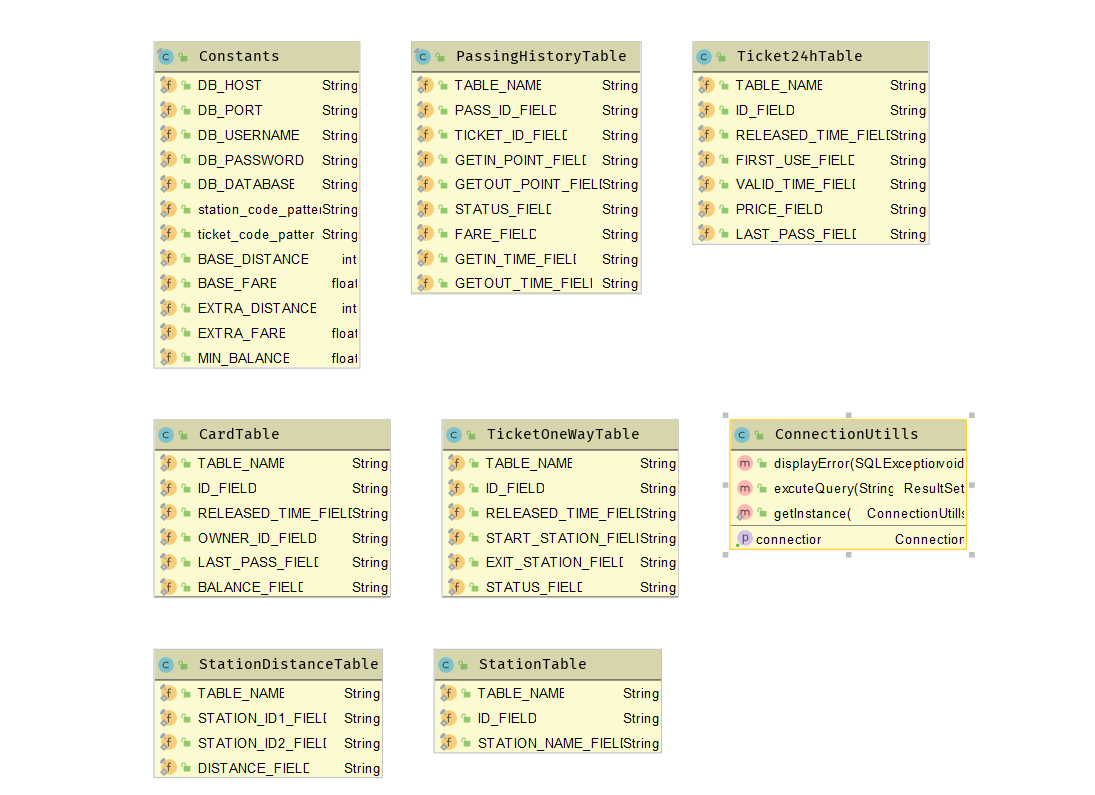


**Class**

|  |  |  |
| --- | --- | --- |
| *#* | *Name* | *Description* |
| 1 | CalculateFare | Interface chứa các phương thức trừu tượng để xử lý việc tính toán phí đi lại |
| 2 | CalculateFarebyDistance | Đại diện cho phương pháp tính phí đi lại dựa trên khoảng cách giữa các nhà ga |

## Package “config”

Mô tả: chứa các class để lưu lại những hằng số được sử dụng trong project

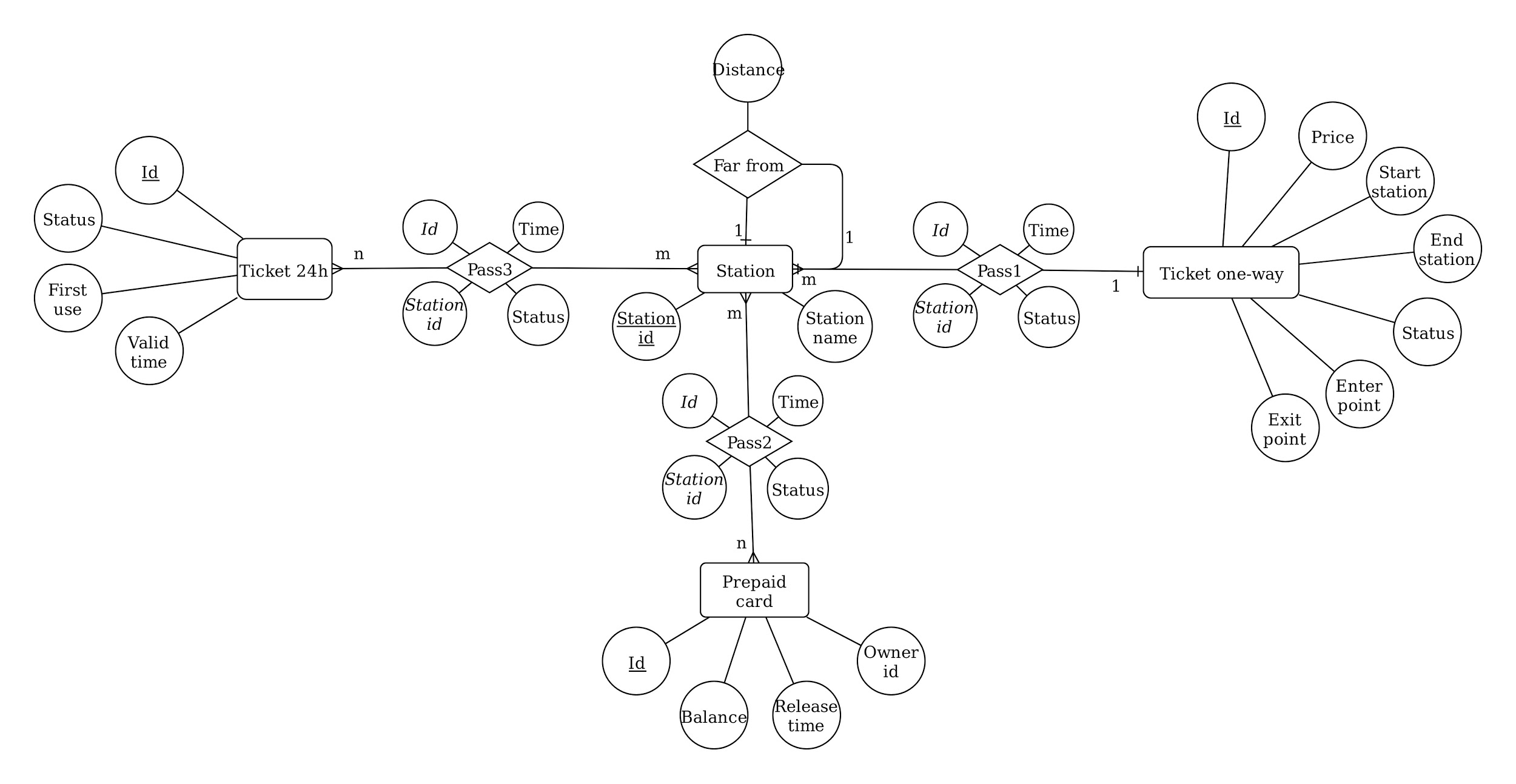


**Class**

|  |  |  |
| --- | --- | --- |
| *#* | *Name* | *Description* |
| 1 | Constants | Lưu các hằng số |
| 2 | PassingHistoryTable | Lưu tên các bảng và tên các trưởng của mỗi bảng trong database |
| 3 | Ticket24hTable |
| 4 | CardTable |
| 5 | TicketOneWayTable |
| 6 | StationDistanceTable |
| 7 | StationTable |
| 8 | ConnectionUtils | Chứa hàm để kết nối tới databse |

# Data Modeling

## Conceptual Data Model



**Thực thể:**

Tiket 24h:

* Id: Là khóa ghi thông tin định danh của vé.
* First use: Thời điểm lần đầu tiên sử dụng vé.
* Valid time: Thời điểm vé hết hạn.
* Status: Trạng thái vé.

Ticket one-way:

* Id: Là khóa ghi định danh của vé.
* Price: Giá tiền mua vé.
* Start station: Ga bắt đầu ( thông tin này được nhập khi mua vé)
* End station: Ga kết thúc ( thông tin này được nhập khi mua vé)
* Status: Trạng thái vé.
* Enter point: Điểm vào ga.
* Exit: point: Điểm ra ga.

Prepaid card:

* Id: Là khóa ghi định danh thẻ.
* Balance: Số dư trong thẻ.
* Release time: Ngày phát hành.
* Owner id: Khóa ngoài, định danh của chủ sở hữu.

Station:

* Station id: Là khóa ghi định danh của nhà ga.
* Name: Tên nhà ga.

Pass:

* *Id*: Khóa ngoài, thông tin định danh vé được sử dụng để đi vào/ra nhà ga.
* *Station id*: Khóa ngoài, thông tin nhà ga được đi qua.
* Time: Thời gian đi qua.
* Status: Trạng thái đi vào hay đi ra.

**Quan hệ**

Ticket one-way --1------------------------<Pass1>---------------------------n--Station

Ticket 24h --n------------------------------<Pass2>---------------------------n--Station

Prepaid card --n--------------------------<Pass3>----------------------------n--Station

Station --1-------------------------------<Far from>----------------------------1--Station

## Logical Data Model

Hệ thống quản trị cơ sở dữ liệu sẽ sử dung: MySQL

Mô tả:

* MySQL là một Relational Database Management System hoạt động theo mô hình client-server
* MySQL được sử dụng để tạo và quản lý các cơ sở dữ liệu (Database) theo hình thức quản lý các mối liên hệ giữa chúng.
* Máy tính cài đặt và chạy phần mềm MySQL được gọi là client (máy khách). Mỗi khi chúng cần truy cập dữ liệu, chúng kết nối tới máy chủ (server) MySQL
* Cách vận hành chính trong môi trường MySQL:
* MySQL tạo ra bảng để lưu trữ dữ liệu, định nghĩa sự liên quan giữa các bảng đó
* Client sẽ gửi yêu cầu SQL bằng một lệnh đặc biệt trên MySQL
* Ứng dụng trên server sẽ phản hồi thông tin và trả về kết quả trên máy client
* Sử dụng MySQL do tính linh hoạt, hiệu năng cao và an toàn của nó.

Các bước thiết kế cơ sở dữ liệu từ sơ đồ được thể liên kết:

* Bước 1: Xác định các mối quan hệ giữa các thực thể:

- One to one

- One to Many

- Many to Many

* Bước 2: Chuẩn hóa cơ sở dữ liệu (1NF, 2NF, 3NF)

- 1NF: + Các thuộc tính của bảng phải là nguyên tố

+ Giá trị của các thuộc tính trên các hàng phải là đơn trị, không chứa nhóm lặp

+ Không có một thuộc tính nào có giá trị có thể tính toán được từ một thuộc tính khác

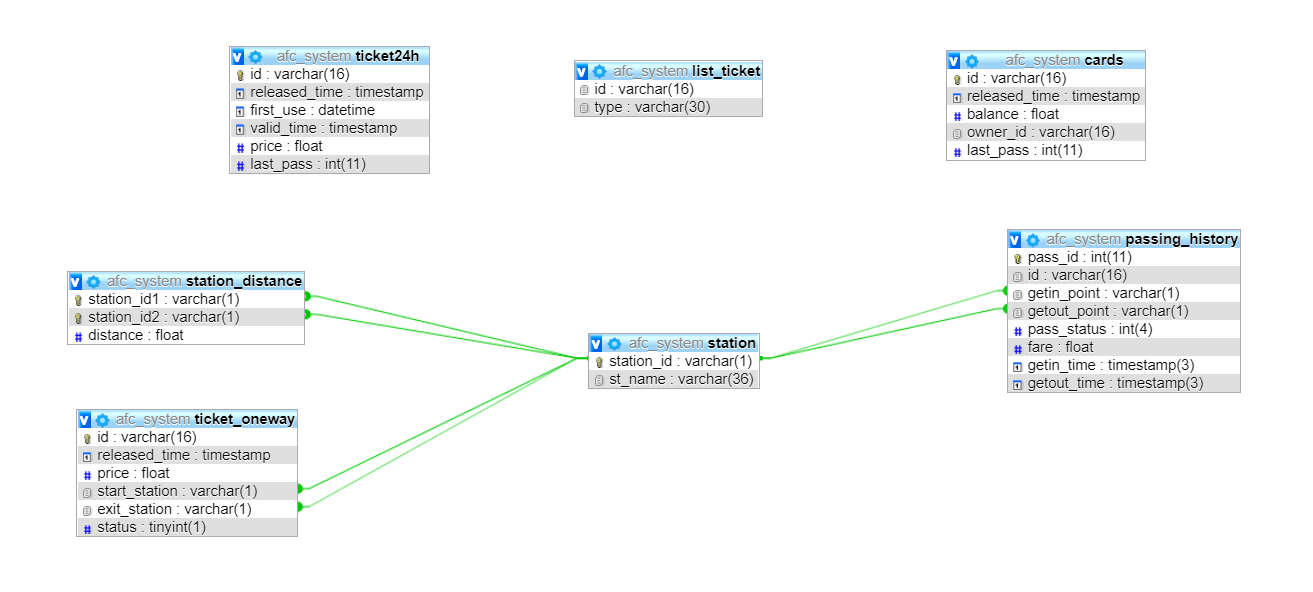
- 2NF:

+ Loại bỏ các thuộc tính không khóa phụ thuộc vào một bộ phận khóa chính và tách ra thành một bảng riêng, khóa chính của bảng là bộ phận của khóa mà chúng phụ thuộc vào.

+ Các thuộc tính còn lại lập thành một quan hệ, khóa chính của nó là khóa chính ban đầu.

- 3NF:

+ Mọi thuộc tính không khóa phụ thuộc bắc cầu vào thuộc tính khóa (nghĩa là tất cả các thuộc tính không khóa phải được suy ra trực tiếp từ thuộc tính khóa)



## Database Design

Bảng lưu thông tin của vé 24h

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *ticket24h* | | | | | | | |
| *#* | *PK* | *FK* | *Column name* | *Data type* | *Default value* | *Mandatory* | *Description* |
| 1 | x |  | id | Varchar(16) |  |  | Mã ID của vé |
| 2 |  |  | released\_time | timestamp | Current\_timestamp() |  | Thời điểm phát hành vé (ngày và giờ) |
| 3 |  |  | first\_use | datetime | NULL |  | Thời điểm sử dụng vé lần đầu |
| 4 |  |  | valid\_time | float | NULL |  | Thời điểm hết hạn sử dụng |
| 5 |  |  | last\_pass | Int(11) | 0 |  | Mã id của lần ra vào gần đây nhất |
| 6 |  |  | price | float | 8.5 |  | Giá vé |

Bảng lưu thông tin của vé một chiều

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *ticket\_oneway* | | | | | | | |
| *#* | *PK* | *FK* | *Column name* | *Data type* | *Default value* | *Mandatory* | *Description* |
| 1 | x |  | id | Varchar(16) |  |  | Mã ID của vé |
| 2 |  |  | released\_time | timestamp | Current\_timestamp() |  | Thời điểm phát hành vé (ngày và giờ) |
| 3 |  |  | start\_station | Varchar(1) |  | notNull | Ga vào mặc định trên vé |
| 4 |  |  | Exit\_station | Varchar(1) |  | notNull | Ga ra mặc định trên vé |
| 5 |  |  | status | Tinyint(1) |  | notNull | Trạng thái đã sử dụng hay chưa |
| 6 |  |  | price | float |  | notNull | Giá vé |

Bảng lưu thông tin của thẻ trả trước

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *cards* | | | | | | | |
| *#* | *PK* | *FK* | *Column name* | *Data type* | *Default value* | *Mandatory* | *Description* |
| 1 | x |  | id | Varchar(16) |  |  | Mã ID của thẻ được tạo khi phát hành |
| 2 |  |  | released\_time | timestamp | Current\_timestamp() |  | Thời điểm phát hành thẻ (ngày và giờ) |
| 3 |  |  | owner\_id | Varchar(16) | 1234567890 abcdef |  | Id của người sở hữu thẻ |
| 4 |  |  | balance | float |  | notnull | Số dư có trong thẻ (lớn hơn hoặc bằng 0) |
| 5 |  |  | Last\_pass | Int(11) | 0 |  | Mã id của lần ra vào gần đây nhất |

Bảng lưu thông tin loại vé/thẻ

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *List\_ticket* | | | | | | | |
| *#* | *PK* | *FK* | *Column name* | *Data type* | *Default value* | *Mandatory* | *Description* |
| 1 | x |  | id | Varchar(16) |  | notnull | Mã ID của thẻ hoặc vé |
| 2 |  |  | type | Varchar(30) |  | notnull | Loại (vé 24h, vé một chiều, thẻ trả trước) |

Bảng lưu thông tin loại các lần vào ra của hành khách

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Passing\_history* | | | | | | | |
| *#* | *PK* | *FK* | *Column name* | *Data type* | *Default value* | *Mandatory* | *Description* |
| 1 | x |  | pass\_id | int(11) |  | notNull | Mã ID một lần vào hoặc ra |
| 2 |  |  | id | Varchar(16) |  | notNull | ID của vé hoặc thẻ dùng để vào ra |
| 3 |  |  | getin\_point | Varchar(1) |  | notNull | ID của ga đi vào |
| 4 |  |  | getout\_point | Varchar(1) | NULL |  | ID của ga đi ra |
| 5 |  |  | pass\_status | int(4) |  | notNull | Tình trạng vào ra (1 là vào, 0 là ra) |
| 6 |  |  | fare | float | NULL |  | Phí đi lại |
| 7 |  |  | getin\_time | Timestamp(3) | Current\_timestamp(3) |  | Thời gian đi vào |
| 8 |  |  | getout\_time | Timestamp(3) | Current\_timestamp(3) |  | Thời gian đi ra |

Bảng lưu thông tin các nhà ga

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *station* | | | | | | | |
| *#* | *PK* | *FK* | *Column name* | *Data type* | *Default value* | *Mandatory* | *Description* |
| 1 | x |  | station\_id | Varchar(1) |  | notNull | Mã ID của nhà ga |
| 2 |  |  | st\_name | Varchar(36) |  | notNull | Tên nhà ga |

Bảng lưu thông tin khoảng cách giưa các nhà ga

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Station\_distance* | | | | | | | |
| *#* | *PK* | *FK* | *Column name* | *Data type* | *Default value* | *Mandatory* | *Description* |
| 1 | x |  | station\_id1 | Varchar(1) |  | notNull | Mã ID của nhà ga thứ nhất |
| 2 | x |  | station\_id2 | Varchar(1) |  | notNull | Mã ID của nhà ga thứ hai |
| 3 |  |  | distance | float |  | notNull | Khoảng cách giữa hai nhà ga |