BỘ GIÁO DỤC VÀ ĐÀO TẠO

**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BỘ MÔN CƠ SỞ DỮ LIỆU

------*------*



**BÁO CÁO DỰ ÁN**

**CƠ SỞ DỮ LIỆU**

**QUẢN LÝ BÃI ĐỖ XE**

**GIÁO VIÊN BỘ MÔN NHÓM SV THỰC HIỆN:**

**GV Nguyễn Hữu Đức Nguyễn Hoài Nam; MSSV: 20225748**

**Nguyễn Đức Mạnh; MSSV:** 20225880

Lê Hồ Quang Huy**; MSSV:** 20225636

**Ngành: Công nghệ thông tin Việt Nhật**

**Tháng 06/2024**

MỤC LỤC

*Trang*

[CHƯƠNG I 1](#_Toc169470140)

[mô tả chi tiết 1](#_Toc169470141)

[CHƯƠNG II 2](#_Toc169470144)

[THIẾT KẾ CƠ SỞ DỮ LIỆU 2](#_Toc169470145)

[CHƯƠNG III 8](#_Toc169470148)

[CÁC CÂU LỆNH SQL 8](#_Toc169470149)

[CHƯƠNG IV 12](#_Toc169470152)

[CÁC CHỨC NĂNG ĐÃ CÀI ĐẶT 12](#_Toc169470153)

[CHƯƠNG V 15](#_Toc169470156)

[KHÓ KHĂN VÀ GIẢI PHÁP 15](#_Toc169470157)

[CHƯƠNG VI 17](#_Toc169470160)

[ĐÁNH GIÁ KẾT QUẢ 17](#_Toc169470161)

[NHIỆM VỤ CỦA CÁC THÀNH VIÊN 18](#_Toc169470164)

CHƯƠNG I:  Mô tả chi tiết

1. ****Giới thiệu****

**1.1 Bối cảnh**

**Một số bãi đỗ xe của các trường đại học đang sử dụng hình thức xử lý ra vào xe đơn giản. Mỗi khách hàng được phát vé khi vào và kiểm tra vé lúc ra. Với hình thức vé thô sơ, không có sự xác thực thông tin lúc ra so với lúc vào, nên đã xảy ra nhiều sự cố như mất vé, trộm cắp. Đồng thời không có hệ thống quản lý để phân bố cho khách hàng vị trí đỗ xe hợp lí.**

**1.2 Mục tiêu**

Với bối cảnh như vậy, chúng tôi thiết kế một hệ thống đơn giản đáp ứng tính tự động tiết kiệm thời gian khi xe ra vào, và đảm bảo an ninh, dễ dàng quản lý khách hàng, nhân viên, bãi đỗ, thống kê doanh thu nhưng cũng đáp ứng về yêu cầu chi phí

1. ****Yêu cầu****

2.1 Người dùng

Có 3 loại người dùng sẽ sử dụng hệ thống: khách hàng, nhân viên và admin.

Khách hàng có hai loại: sinh viên và khách vãng lai. Khách hàng có thể sử dụng dịch vụ đỗ xe, truy cập một số thông tin cần thiết.

Nhân viên chịu trách nhiệm cho *quá trình xe ra vào,* đảm bảo thông tin khách hàng gửi đến hệ thống cũng như truyền thông tin phản hồi từ hệ thống cho khách hàng.

Admin có khả năng điều chỉnh thay đổi thông tin khách hàng, nhân viên, bãi đỗ xe, chỗ đỗ xe, cũng như thiết kế chỉnh sửa database.

2.2 Chức năng

a. Khách hàng

* Chức năng chung

Khi khách hàng gửi xe vào bãi, lưu lại thông tin xe (màu, biển số, loại xe), chỗ đỗ xe và thời gian ra vào.

* Chức năng riêng đối với từng khách hàng

\* Đối với sinh viên

1. Chỉ có thể gửi một xe bằng thẻ sinh viên trong cùng một thời điểm ở tất cả các bãi xe của hệ thống.
2. Có tài khoản để đăng nhập vào giao diện sử dụng các chức năng
   * + Số dư trong tài khoản
       - Số tiền còn lại sinh viên có thể dùng để gửi xe.
     + Xem thông tin vị trí đang đỗ xe
     + Nạp tiền
       - Sinh viên có khả năng nạp tiền tích lũy vào tài khoản để sử dụng cho việc gửi xe.
     + Xem lịch sử giao dịch nạp tiền và trả tiền gửi xe
     + Đổi mật khẩu
3. Đăng ký tài khoản sử dụng bãi đỗ xe thông qua chức năng Sign Up

\* Đối với khách vãng lai

Khi khách vãng lai vào bãi, khách sẽ được cấp phát một mã vé. Vé này được sử dụng để kiểm tra khi khách ra khỏi bãi.

Lưu lại thông tin xe ra vào bãi.

b. Nhân viên

1. Có tài khoản đăng nhập vào hệ thống để làm việc

* Nhập thông tin xe vào và khách hàng, ra gửi cho hệ thống
* Tiếp nhận phản hồi từ hệ thống để thông báo cho khách hàng
  + Từ chối xe vào khi sinh viên không còn đủ số dư, hoặc sinh viễn đã gửi 1 xe trong bãi.
  + Chỗ khách hàng được phép đỗ
  + Từ chối xe ra trong trường hợp sai vé, xe lấy ra không phải xe của khách hàng

2. Có thể ứng tuyển làm việc tại bãi đỗ xe thông qua Sign Up

c. Admin

1. Có khả năng thêm xóa sửa Sinh viên

2. Có khả năng thêm xóa sửa Nhân viên, duyệt đơn xin việc làm của Nhân viên

3. Có khả năng thêm xóa sửa Bãi đỗ xe

4. Có khả năng thêm xóa sửa Chỗ đỗ xe

5. Có khả năng truy vấn doanh thu theo (nhiều) ngày, (nhiều) tháng và trong năm vừa qua.

6. Có khả năng tương tác với Database thông qua giao diện dòng lệnh đơn giản (sử dụng SQL trực tiếp)

2.3 Yêu cầu hệ thống

Hệ thống có khả năng đáp ứng các xử lý khi xe vào, cũng như xe ra.

a. Khi xe vào

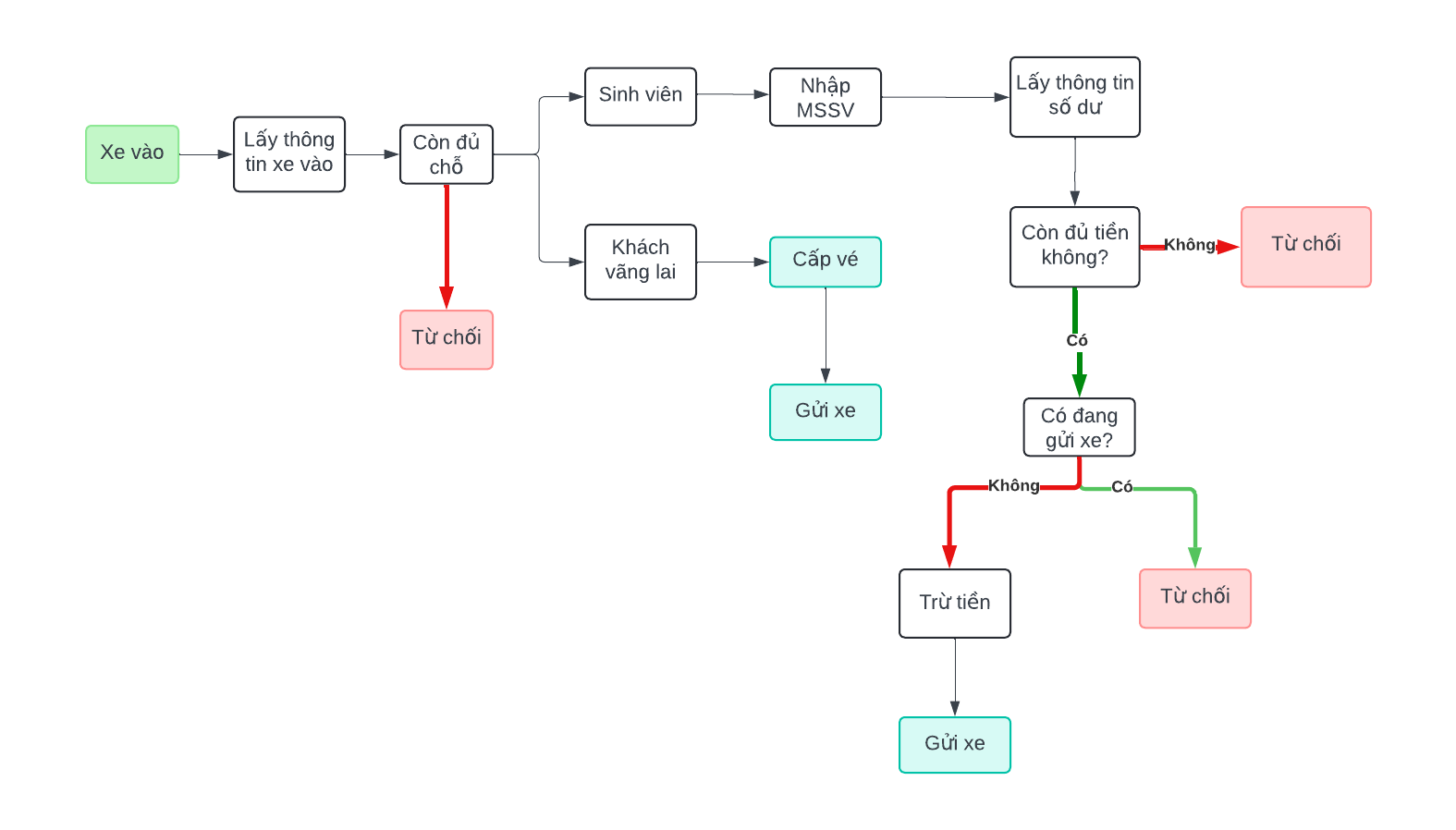


Figure 1. Quy trình lúc xe vào

Khi xe vào một bãi đỗ, nhập thông tin xe vào: màu xe, biển số, loại xe. Hệ thống kiểm tra xem còn đủ chỗ cho loại xe đó không. Nếu không còn chỗ, gửi thông báo từ chối cho Nhân viên. Nếu còn đủ chỗ, thì với mỗi loại khách hàng hệ thống có cách xử lý khác nhau.

Đối với khách vãng lai, hệ thống cấp mã vẽ cho khách hàng, rồi thực hiện quy trình *Gửi xe*.

Đối với sinh viên, sau khi gửi thông tin mã số sinh viên cho hệ thống, hệ thống kiểm tra số dư có còn đủ tiền cho loại xe này không. Nếu không còn đủ tiền cho giao dịch, gửi thông báo từ chối vào bãi. Nếu có thì hệ thống sẽ kiểm tra sinh viên này đã gửi xe trong bãi hay chưa. Nếu đã gửi thì sẽ từ chối vào bãi vì mỗi sinh viên chỉ được phép gửi 1 xe trong tất cả các bãi của trường trong một thời điểm. Nếu chưa gửi, thực hiện quy trình Gửi xe.

b. Gửi xe

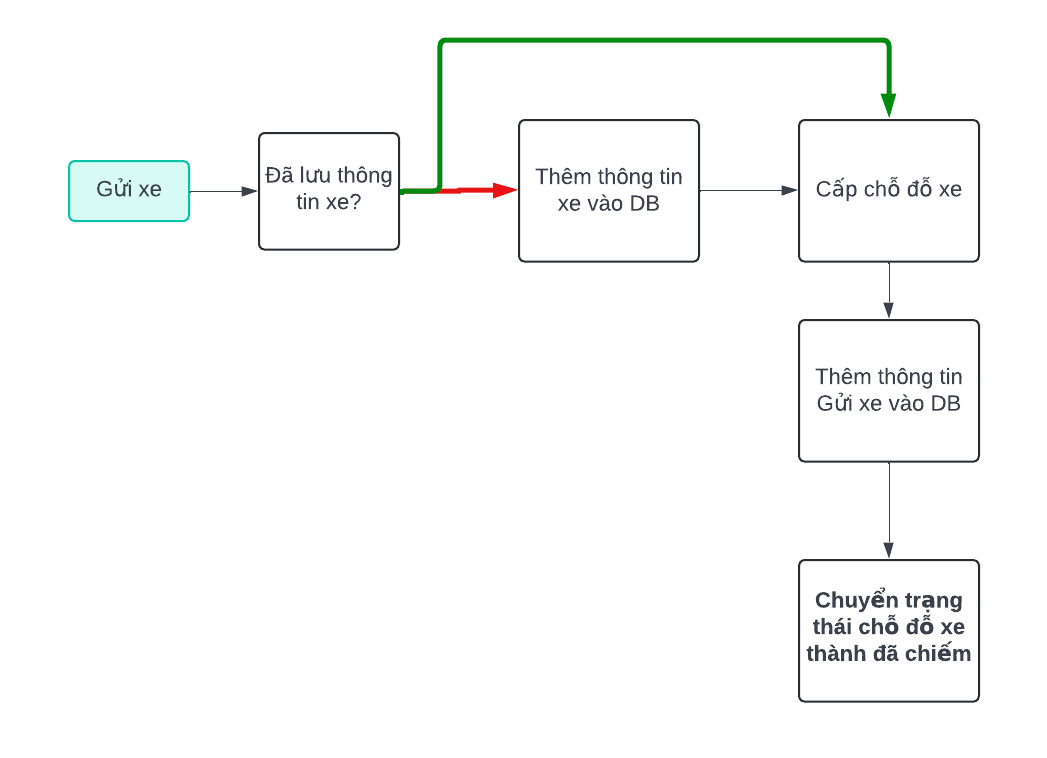
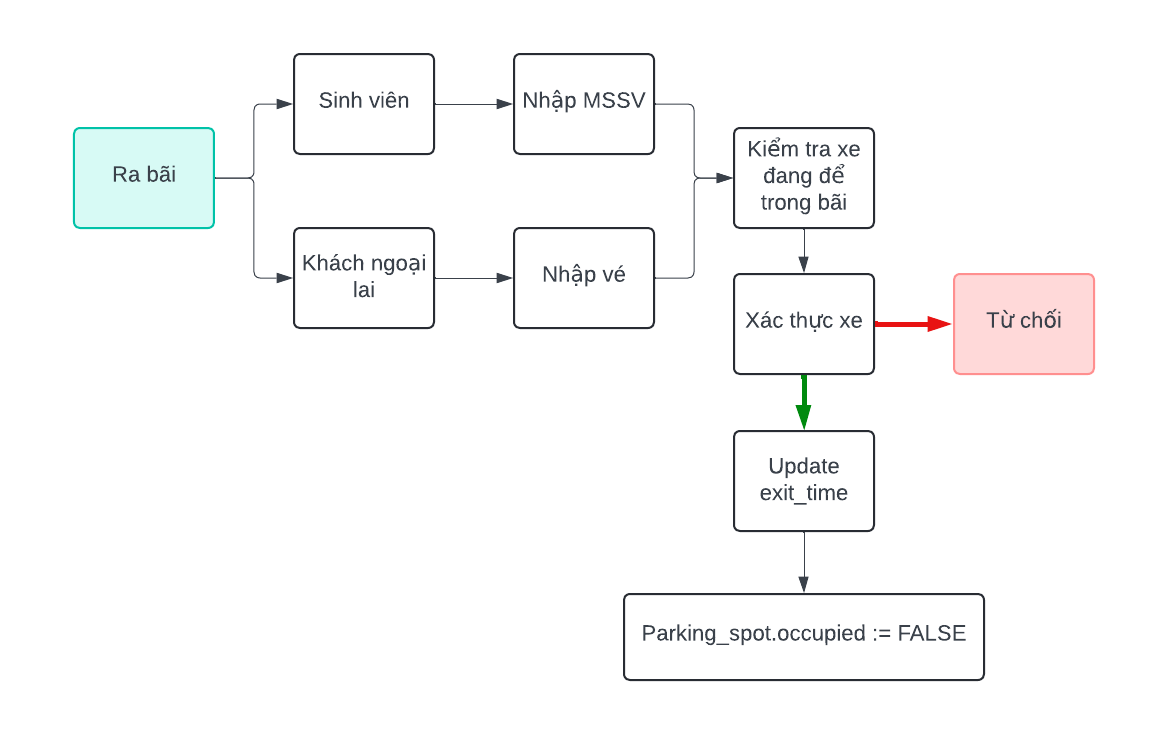


Figure 2. Quy trình Gửi xe

Khi khách hàng được phép vào bãi, thông tin xe của khách hàng sẽ được kiểm tra đã lưu trong hệ thống hay chưa. Để đảm bảo không có sự dư thừa dữ liệu do lưu lại thông tin xe đã tồn tại của sinh viên, chỉ thêm thông tin xe mỗi lần sinh viên đi xe chưa từng đi. Sau khi xử lý xong thông tin, cấp chỗ đỗ xe phù hợp cho loại xe đó. Lưu thông tin gửi đỗ vào khách hàng. Đồng thời chỗ mà xe được cấp chỗ chuyển sang trạng thái đã chiếm.

c. Xe ra

Khi xe ra, sau khi nhập thông tin để xác thực xe đang gửi trong bãi. Sinh viên quét hay nhập mã số sinh viên, khách vãng lai cấp mã vé cho nhân viên. Hệ thống quét lại thông tin xe ra và xác thực xem có trùng khớp với xe đang gửi trong bãi hay không. Nếu được cho phép ra bãi, cập nhật thời gian xe ra của khách hàng và trả lại trạng thái **còn trống** cho chỗ đỗ xe đó.



d. Lưu lịch sử thanh toán

Mỗi lần sinh viên nạp tiền, lưu lịch sử nạp tiền.

Mỗi lần khách hàng trả tiền gửi xe, lưu lịch sử trả tiền.

### II. Công nghệ sử dụng

Hệ thống được phát triển sử dụng các công nghệ và thư viện sau:

* **PostgreSQL**: Hệ quản trị cơ sở dữ liệu.
* **Python**: Ngôn ngữ lập trình chính sử dụng để triển khai các chức năng.
* **psycopg2**: Thư viện Python để kết nối và tương tác với PostgreSQL.
* **tabulate**: Thư viện hỗ trợ việc hiển thị dữ liệu dạng bảng.
* **os**: Thư viện hỗ trợ các thao tác hệ thống.
* **và một số thư viện hỗ trợ khác**.

III. Quy trình triển khai

#### 5.1 Xác định Yêu cầu

* **Thu thập yêu cầu**: Tiến hành thu thập yêu cầu từ các bên liên quan (nhân viên quản lý bãi đỗ, sinh viên, bộ phận IT) để xác định rõ các yêu cầu và mong muốn.
* **Xác định yêu cầu chức năng**: Xác định các yêu cầu cụ thể về chức năng của hệ thống, bao gồm việc quản lý ra vào bãi đỗ, quản lý thông tin nhân viên và sinh viên, và quản lý thông tin chỗ đỗ xe.

#### 5.2 Thiết kế Hệ thống

* **Thiết kế ERD**: Xác định các thực thể và mối quan hệ giữa chúng trong hệ thống.
* **Thiết kế Cơ sở dữ liệu**: Dựa trên ERD, tạo các bảng và cấu trúc dữ liệu trong PostgreSQL.

#### 5.3 Triển khai Hệ thống

* **Tạo cơ sở dữ liệu**: Sử dụng các lệnh SQL để tạo các bảng trong PostgreSQL theo thiết kế ERD.
* **Phát triển các chức năng bằng Python**: Sử dụng thư viện psycopg2 để kết nối và thao tác với cơ sở dữ liệu.

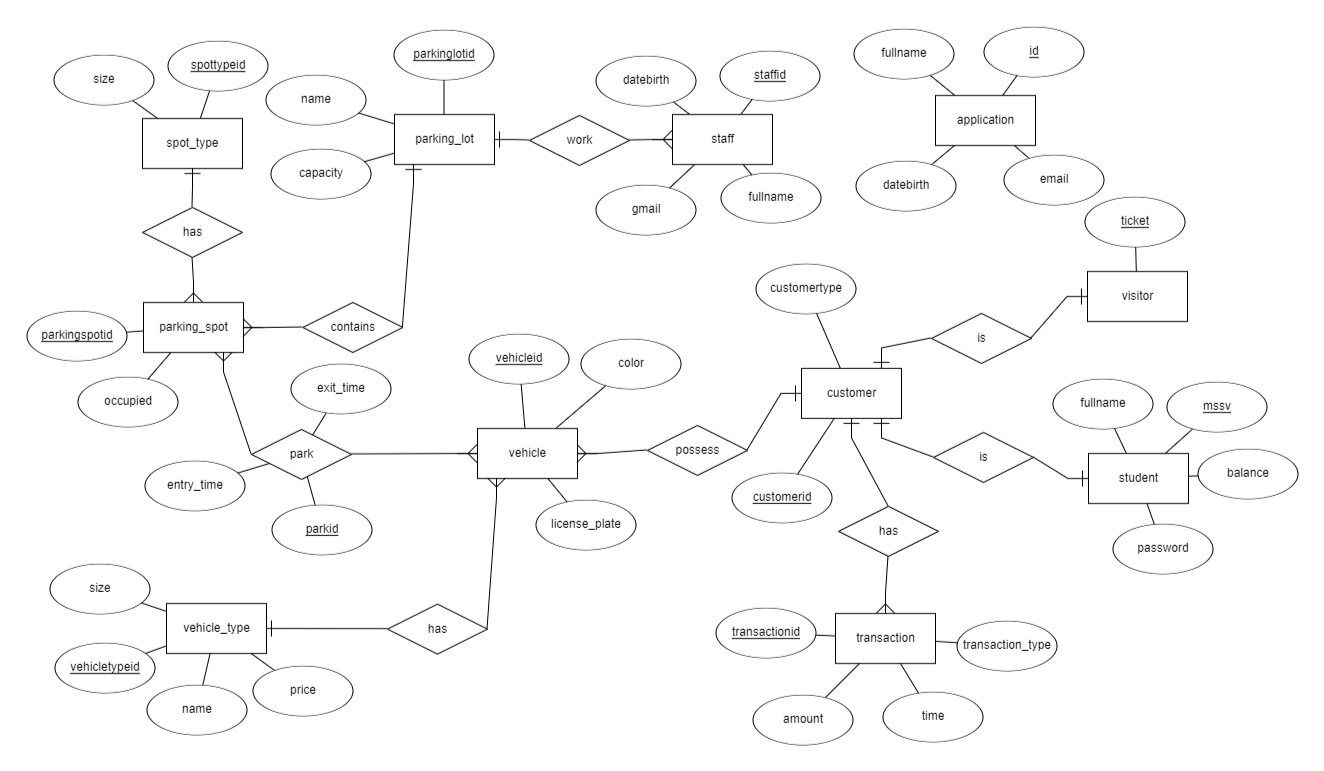
#### 5.5 Lập báo cáo

* **Tổng hợp kết quả**: Thu thập kết quả từ các giai đoạn thiết kế, triển khai.
* **Viết báo cáo**: Trình bày chi tiết quá trình thực hiện, kết quả đạt được, các vấn đề gặp phải và hướng giải quyết.
* **Đề xuất cải tiến**: Đưa ra các đề xuất cải tiến cho hệ thống trong tương lai.

CHƯƠNG II

THIẾT KẾ CƠ SỞ DỮ LIỆU

Sơ đồ thực thể liên kết ( E RD):



1. **Mô tả Sơ đồ ER**:

* Sơ đồ ER (Entity-Relationship) được sử dụng để mô tả các thực thể trong hệ thống quản lý bãi đậu xe và mối quan hệ giữa chúng. Dưới đây là các thực thể chính và mối quan hệ giữa chúng trong hệ thống.

1. **Thực thể và mối quan hệ**

 **PARKING\_LOT (Bãi đậu xe)**

* **Thuộc tính**:
  + id: Khóa chính, định danh duy nhất cho mỗi bãi đậu xe.
  + name: Tên bãi đậu xe.
  + capacity: Sức chứa của bãi đậu xe.
* **Quan hệ**:
  + Quản lý bởi STAFF.

 **STAFF (Nhân viên)**

* **Thuộc tính**:
  + StaffID: Khóa chính, định danh duy nhất cho mỗi nhân viên.
  + Full Name: Tên đầy đủ của nhân viên.
  + Password: Mật khẩu truy cập hệ thống.
  + Parking\_Lot\_ID: Khóa ngoại liên kết với PARKING\_LOT.

 **SPOT\_TYPE (Loại chỗ đậu)**

* **Thuộc tính**:
  + spotTypeID: Khóa chính, định danh loại chỗ đậu.
  + vehicleType: Loại xe phù hợp với chỗ đậu.
  + isPrivileged: Chỗ đậu ưu tiên.

 **PARKING\_SPOT (Chỗ đậu xe)**

* **Thuộc tính**:
  + id: Khóa chính, định danh chỗ đậu xe.
  + spot\_type\_id: Khóa ngoại liên kết với SPOT\_TYPE.
  + parking\_lot\_id: Khóa ngoại liên kết với PARKING\_LOT.
  + occupied: Trạng thái của chỗ đậu (đang đậu/không).

 **CUSTOMER (Khách hàng)**

* **Thuộc tính**:
  + CustomerID: Khóa chính, định danh khách hàng.
  + CustomerType: Loại khách hàng (học sinh/khách).

 **VISITOR (Khách)**

* **Thuộc tính**:
  + CustomerID: Khóa chính, định danh khách hàng (liên kết với CUSTOMER).
  + CardID: ID thẻ khách.

 **STUDENTS (Học sinh)**

* **Thuộc tính**:
  + CustomerID: Khóa chính, định danh học sinh (liên kết với CUSTOMER).
  + Full Name: Tên đầy đủ.
  + MSSV: Mã số sinh viên.
  + Balance: Số dư tài khoản.
  + Password: Mật khẩu truy cập hệ thống.

 **VEHICLE (Xe)**

* **Thuộc tính**:
  + VehicleID: Khóa chính, định danh xe.
  + typeID: Khóa ngoại liên kết với VEHICLE\_TYPE.
  + license\_plate: Biển số xe.
  + color: Màu xe.
  + CustomerID: Khóa ngoại liên kết với CUSTOMER.

 **VEHICLE\_TYPE (Loại xe)**

* **Thuộc tính**:
  + typeID: Khóa chính, định danh loại xe.
  + name: Tên loại xe.
  + price: Giá loại xe.

 **PARK (Đậu xe)**

* **Thuộc tính**:
  + ParkID: Khóa chính, định danh lượt đậu xe.
  + VehicleID: Khóa ngoại liên kết với VEHICLE.
  + Parking\_Slot\_ID: Khóa ngoại liên kết với PARKING\_SPOT.
  + entry\_time: Thời gian vào.
  + exit\_time: Thời gian ra.

 **DepositTransaction (Giao dịch nạp tiền)**

* **Thuộc tính**:
  + transaction\_id: Khóa chính, định danh giao dịch.
  + MSSV: Khóa ngoại liên kết với STUDENTS.
  + amount: Số tiền giao dịch.
  + deposit\_date: Ngày giao dịch.
  + transaction\_type: Loại giao dịch.

####  Quan hệ giữa các thực thể

* **PARKING\_LOT** có thể chứa nhiều **PARKING\_SPOT**.
* **SPOT\_TYPE** xác định các loại **PARKING\_SPOT**.
* **CUSTOMER** có thể là **VISITOR** hoặc **STUDENTS**.
* **CUSTOMER** có thể sở hữu nhiều **VEHICLE**.
* **VEHICLE** có loại xe xác định bởi **VEHICLE\_TYPE**.
* **PARK** liên kết **VEHICLE** với **PARKING\_SPOT** tại một thời điểm cụ thể.
* **STUDENTS** có thể thực hiện nhiều **DepositTransaction**.

### Lược đồ cơ sở dữ liệu

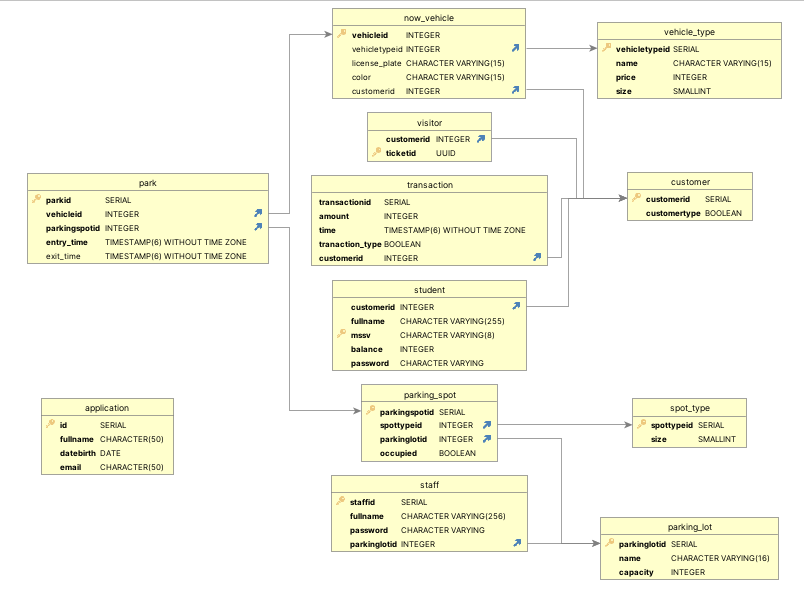


Figure 3. Lược đồ CSDL

CHƯƠNG III

CÁC CÂU LỆNH SQL

**Các câu lệnh SQL từ mã nguồn**:

* 1. **Thêm người dùng mới**:

cur.execute("INSERT INTO users (username, password) VALUES (?, ?)", (username, password))

* 1. **Đăng nhập**:

cur.execute("SELECT \* FROM users WHERE username = ? AND password = ?", (username, password))

* 1. **Thêm xe vào hệ thống**:

cur.execute("INSERT INTO vehicles (user\_id, license\_plate, vehicle\_type) VALUES (?, ?, ?)", (user\_id, license\_plate, vehicle\_type))

* 1. **Đặt chỗ đậu xe**:

cur.execute("INSERT INTO reservations (user\_id, spot\_id, start\_time, end\_time) VALUES (?, ?, ?, ?)", (user\_id, spot\_id, start\_time, end\_time))

* 1. **Tính toán doanh thu**:

cur.execute("INSERT INTO revenues (reservation\_id, amount) VALUES (?, ?)", (reservation\_id, amount))

* 1. **Cập nhật thông tin người dùng**:

cur.execute("UPDATE users SET password = ? WHERE username = ?", (new\_password, username))

* 1. **Xóa người dùng**:

cur.execute("DELETE FROM users WHERE username = ?", (username,))

* 1. **Tìm kiếm xe theo biển số**:

cur.execute("SELECT \* FROM vehicles WHERE license\_plate = ?", (license\_plate,))

* 1. **Danh sách tất cả các chỗ đậu xe**:

cur.execute("SELECT \* FROM parking\_spots")

* 1. **Kiểm tra tình trạng chỗ đậu xe**:

cur.execute("SELECT \* FROM reservations WHERE spot\_id = ? AND start\_time <= ? AND end\_time >= ?", (spot\_id, current\_time, current\_time))

* 1. **Tổng doanh thu trong một khoảng thời gian**:

cur.execute("SELECT SUM(amount) FROM revenues WHERE reservation\_id IN (SELECT reservation\_id FROM reservations WHERE start\_time >= ? AND end\_time <= ?)", (start\_time, end\_time))

* 1. **Thông tin người dùng theo ID**:

cur.execute("SELECT \* FROM users WHERE user\_id = ?", (user\_id,))

* 1. **Thông tin xe theo ID người dùng**:

cur.execute("SELECT \* FROM vehicles WHERE user\_id = ?", (user\_id,))

* 1. **Thông tin chỗ đậu xe theo ID**:

cur.execute("SELECT \* FROM parking\_spots WHERE spot\_id = ?", (spot\_id,))

* 1. **Danh sách tất cả các đặt chỗ của người dùng**:

cur.execute("SELECT \* FROM reservations WHERE user\_id = ?", (user\_id,))

* 1. **Cập nhật thông tin xe**:

cur.execute("UPDATE vehicles SET license\_plate = ?, vehicle\_type = ? WHERE vehicle\_id = ?", (license\_plate, vehicle\_type, vehicle\_id))

* 1. **Xóa thông tin xe**:

cur.execute("DELETE FROM vehicles WHERE vehicle\_id = ?", (vehicle\_id,))

* 1. **Thêm chỗ đậu xe mới**:

cur.execute("INSERT INTO parking\_spots (location, size) VALUES (?, ?)", (location, size))

* 1. **Cập nhật thông tin chỗ đậu xe**:

cur.execute("UPDATE parking\_spots SET location = ?, size = ? WHERE spot\_id = ?", (location, size, spot\_id))

* 1. **Xóa chỗ đậu xe**:

cur.execute("DELETE FROM parking\_spots WHERE spot\_id = ?", (spot\_id,))

* 1. **Kiểm tra doanh thu của một đặt chỗ cụ thể**:

cur.execute("SELECT amount FROM revenues WHERE reservation\_id = ?", (reservation\_id,))

* 1. **Cập nhật doanh thu của một đặt chỗ**:

cur.execute("UPDATE revenues SET amount = ? WHERE reservation\_id = ?", (amount, reservation\_id))

* 1. **Xóa doanh thu của một đặt chỗ**:

cur.execute("DELETE FROM revenues WHERE reservation\_id = ?", (reservation\_id,))

* 1. **Danh sách tất cả các doanh thu**:

cur.execute("SELECT \* FROM revenues")

* 1. **Kiểm tra thông tin đặt chỗ theo ID**:

cur.execute("SELECT \* FROM reservations WHERE reservation\_id = ?", (reservation\_id,))

* 1. **Cập nhật thông tin đặt chỗ**:

cur.execute("UPDATE reservations SET spot\_id = ?, start\_time = ?, end\_time = ? WHERE reservation\_id = ?", (spot\_id, start\_time, end\_time, reservation\_id))

* 1. **Xóa đặt chỗ**:

cur.execute("DELETE FROM reservations WHERE reservation\_id = ?", (reservation\_id,))

* 1. **Kiểm tra tổng số lượng người dùng**:

cur.execute("SELECT COUNT(\*) FROM users")

* 1. **Kiểm tra tổng số lượng xe**:

cur.execute("SELECT COUNT(\*) FROM vehicles")

* 1. **Kiểm tra tổng số lượng chỗ đậu xe**:

cur.execute("SELECT COUNT(\*) FROM parking\_spots")

CHƯƠNG IV

CÁC CHỨC NĂNG ĐÃ CÀI ĐẶT

**1. Đăng ký và quản lý tài khoản người dùng**

* **Đăng ký người dùng**: Người dùng có thể tạo tài khoản mới bằng cách cung cấp tên đăng nhập và mật khẩu.

cur.execute("INSERT INTO users (username, password) VALUES (?, ?)", (username, password))

* **Đăng nhập**: Người dùng có thể đăng nhập vào hệ thống bằng tên đăng nhập và mật khẩu.

cur.execute("SELECT \* FROM users WHERE username = ? AND password = ?", (username, password))

* **Cập nhật thông tin người dùng**: Người dùng có thể cập nhật mật khẩu.

cur.execute("UPDATE users SET password = ? WHERE username = ?", (new\_password, username))

**2. Quản lý thông tin xe**

* **Thêm xe mới**: Người dùng có thể thêm thông tin xe của mình vào hệ thống.

cur.execute("INSERT INTO vehicles (user\_id, license\_plate, vehicle\_type) VALUES (?, ?, ?)", (user\_id, license\_plate, vehicle\_type))

* **Cập nhật thông tin xe**: Người dùng có thể cập nhật thông tin xe.

cur.execute("UPDATE vehicles SET license\_plate = ?, vehicle\_type = ? WHERE vehicle\_id = ?", (license\_plate, vehicle\_type, vehicle\_id))

* **Xóa xe**: Người dùng có thể xóa thông tin xe.

cur.execute("DELETE FROM vehicles WHERE vehicle\_id = ?", (vehicle\_id,))

**3. Đặt chỗ đậu xe**

* **Đặt chỗ**: Người dùng có thể đặt chỗ đậu xe dựa trên thông tin về bãi đậu xe và loại xe.

cur.execute("INSERT INTO reservations (user\_id, spot\_id, start\_time, end\_time) VALUES (?, ?, ?, ?)", (user\_id, spot\_id, start\_time, end\_time))

* **Cập nhật thông tin đặt chỗ**: Người dùng có thể cập nhật thông tin đặt chỗ.

cur.execute("UPDATE reservations SET spot\_id = ?, start\_time = ?, end\_time = ? WHERE reservation\_id = ?", (spot\_id, start\_time, end\_time, reservation\_id))

* **Xóa đặt chỗ**: Người dùng có thể xóa thông tin đặt chỗ.

cur.execute("DELETE FROM reservations WHERE reservation\_id = ?", (reservation\_id,))

**4. Quản lý doanh thu**

* **Tính toán và thêm doanh thu**: Hệ thống tính toán và thêm doanh thu từ các lượt đặt chỗ.

cur.execute("INSERT INTO revenues (reservation\_id, amount) VALUES (?, ?)", (reservation\_id, amount))

* **Cập nhật doanh thu**: Cập nhật thông tin doanh thu.

cur.execute("UPDATE revenues SET amount = ? WHERE reservation\_id = ?", (amount, reservation\_id))

* **Xóa doanh thu**: Xóa thông tin doanh thu.

cur.execute("DELETE FROM revenues WHERE reservation\_id = ?", (reservation\_id,))

**5. Báo cáo và thống kê**

* **Tổng doanh thu trong khoảng thời gian**: Tính toán tổng doanh thu trong một khoảng thời gian cụ thể.

cur.execute("SELECT SUM(amount) FROM revenues WHERE reservation\_id IN (SELECT reservation\_id FROM reservations WHERE start\_time >= ? AND end\_time <= ?)", (start\_time, end\_time))

* **Danh sách tất cả các chỗ đậu xe**: Truy vấn danh sách tất cả các chỗ đậu xe hiện có.

cur.execute("SELECT \* FROM parking\_spots")

CHƯƠNG V

KHÓ KHĂN VÀ GIẢI PHÁP

#### Khó khăn 1: Xác định và thống nhất các yêu cầu của hệ thống

* **Mô tả:** Một trong những thách thức đầu tiên mà nhóm gặp phải là xác định và thống nhất các yêu cầu của hệ thống. Mỗi thành viên có những quan điểm và ý tưởng riêng, dẫn đến khó khăn trong việc đạt được sự đồng thuận về các chức năng cần thiết và cách thức triển khai.
* **Giải pháp:** Để giải quyết vấn đề này, nhóm em đã tổ chức nhiều cuộc họp để thảo luận và lắng nghe ý kiến của từng thành viên. Sau đó, cả nhóm đã sử dụng các công cụ hỗ trợ như ChatGPT và tài liệu môn học để ghi lại và thống nhất các ý tưởng của mọi người một cách chi tiết và rõ ràng.

#### Khó khăn 2: Thiết kế cơ sở dữ liệu

* **Mô tả:** Thiết kế cơ sở dữ liệu là một phần quan trọng và phức tạp của dự án. Cả nhóm gặp khó khăn trong việc tạo ra một lược đồ ER hoàn chỉnh và tối ưu hóa các bảng để đảm bảo tính toàn vẹn và hiệu suất của cơ sở dữ liệu.
* **Giải pháp:** Nhóm đã tham khảo nhiều tài liệu và hướng dẫn về thiết kế cơ sở dữ liệu, đồng thời nhờ sự hỗ trợ từ giảng viên và các chuyên gia. Nhóm cũng đã tiến hành kiểm tra và sửa lỗi lược đồ ER nhiều lần trước khi chốt phương án cuối cùng.

#### Khó khăn 3: Viết và tối ưu hóa các câu lệnh SQL

* **Mô tả:** Viết các câu lệnh SQL phức tạp và tối ưu hóa chúng để đảm bảo hiệu suất cao là một thách thức không nhỏ. Một số câu lệnh ban đầu chạy rất chậm hoặc không trả về kết quả đúng.
* **Giải pháp:** Nhóm đã tiến hành kiểm tra từng câu lệnh SQL một cách cẩn thận và sử dụng ChatGPT hỗ trợ để xác định các điểm yếu và đưa ra gợi ý. Dựa vào kết quả phân tích, nhóm đã điều chỉnh và tối ưu hóa các câu lệnh SQL để đạt được hiệu suất tốt nhất.

#### Khó khăn 4: Triển khai các chức năng đã thiết kế

* **Mô tả:** Một số chức năng được xác định ban đầu khá phức tạp và yêu cầu nhiều thời gian và công sức để triển khai. Điều này gây ra áp lực lớn về thời gian cho nhóm.
* **Giải pháp:** Nhóm đã phân chia công việc một cách hợp lý và ưu tiên triển khai các chức năng quan trọng trước. Nhóm cũng áp dụng các phương pháp lập trình theo từng bước nhỏ (incremental development) để có thể kiểm tra và sửa lỗi kịp thời.

#### Khó khăn 5: Phối hợp làm việc nhóm

* **Mô tả:** Phối hợp làm việc giữa các thành viên trong nhóm là một thách thức, đặc biệt khi có sự khác biệt về thời gian rảnh và sở trường code.
* **Giải pháp:** Nhóm đã sử dụng các công cụ quản lý dự án như Trello và Slack để theo dõi tiến độ công việc và giao tiếp hiệu quả. Nhóm cũng đã đặt ra các cuộc họp để cập nhật tiến độ và giải quyết các vấn đề phát sinh.

CHƯƠNG VI

ĐÁNH GIÁ KẾT QUẢ

**Ưu điểm:**

* **Hoạt động ổn định:** Hệ thống quản lý bãi đỗ xe HUST đã hoàn thành và hoạt động ổn định, thực hiện đúng các chức năng đã đề ra như quản lý tài khoản, xe, bãi đỗ, và theo dõi doanh thu.
* **Giao diện dễ sử dụng:** Giao diện người dùng thân thiện và rõ ràng, giúp người dùng dễ dàng thao tác và sử dụng hệ thống.
* **Hiệu quả quản lý:** Hệ thống giúp quản lý thông tin xe và bãi đỗ một cách hiệu quả, tiết kiệm thời gian và công sức so với phương pháp truyền thống.

**Nhược điểm:**

* **Tối ưu hóa chưa hoàn thiện:** Một số chức năng và câu lệnh SQL chưa được tối ưu hóa hoàn toàn, gây ra một số vấn đề về hiệu suất khi xử lý dữ liệu lớn.
* **Chức năng hạn chế:** Hệ thống mới chỉ cài đặt một số chức năng cơ bản, cần mở rộng và phát triển thêm các chức năng nâng cao như báo cáo chi tiết, phân tích dữ liệu nâng cao.
* **Giao diện chưa hoàn thiện:** Mặc dù giao diện dễ sử dụng, nhưng còn đơn giản và cần cải tiến về mặt thẩm mỹ để thu hút người dùng hơn.

NHIỆM VỤ CỦA CÁC THÀNH VIÊN

**Nguyễn Đức Mạnh:**

 **Phân tích nghiệp vụ:**

* Nghiên cứu và phân tích các yêu cầu nghiệp vụ của hệ thống quản lý bãi đỗ xe, đảm bảo hiểu rõ các chức năng cần thiết và cách thức hoạt động của hệ thống.
* **Ví dụ:** Xác định các yêu cầu như đăng ký tài khoản, quản lý thông tin xe, và tính toán doanh thu.

 **Thiết kế ERD:**

* Thiết kế lược đồ ER (Entity-Relationship Diagram) để biểu diễn mối quan hệ giữa các thực thể trong cơ sở dữ liệu.
* **Ví dụ:** Tạo lược đồ ER để mô tả các bảng account, car, parking\_lot, và revenue cùng các mối quan hệ của chúng.

 **Tạo cơ sở dữ liệu:**

* Triển khai cơ sở dữ liệu theo thiết kế ERD bằng cách tạo các bảng và định nghĩa các ràng buộc trong SQL.

 **Viết trigger, function, procedure:**

* Phát triển các trigger, function, và procedure để tự động hóa các tác vụ và nâng cao hiệu suất của hệ thống.

 **Làm project:**

* Tham gia vào các giai đoạn chính của dự án, từ phân tích yêu cầu, thiết kế cơ sở dữ liệu, đến phát triển các chức năng quan trọng của hệ thống.
* **Ví dụ:** Phát triển chức năng đăng ký và đăng nhập tài khoản, quản lý xe và bãi đỗ, theo dõi và tính toán doanh thu.

**Nguyễn Hoài Nam:**

 **Hoàn thiện báo cáo và slide thuyết trình:**

* Tạo và hoàn thiện báo cáo dự án chi tiết, bao gồm các phần như mô tả chủ đề, thiết kế cơ sở dữ liệu, và các chức năng đã triển khai.
* Thiết kế slide thuyết trình để trình bày kết quả dự án trước lớp hoặc hội đồng.

 **Hoàn thiện phần quản lý nhân viên, học sinh và bãi đỗ xe:**

* Phát triển các chức năng quản lý nhân viên và học sinh, bao gồm thêm, sửa, xóa, và truy vấn thông tin.
* Tối ưu hóa và hoàn thiện các chức năng liên quan đến quản lý bãi đỗ xe.

 **Tổng hợp các ý kiến để đưa ra hướng làm project:**

* Thu thập và tổng hợp các ý kiến từ các thành viên trong nhóm để định hướng phát triển dự án.
* Đảm bảo rằng tất cả các yêu cầu và chức năng cần thiết đều được thảo luận và lên kế hoạch triển khai hợp lý.

**Lê Hồ Quang Huy:**

 **Kiểm tra cơ sở dữ liệu:** Double check toàn bộ cơ sở dữ liệu để đảm bảo tính chính xác và toàn vẹn dữ liệu.

 **Tối ưu hóa ERD:** Cải tiến lược đồ ERD (Entity-Relationship Diagram) để đảm bảo hiệu quả và tối ưu hóa.

 **Viết trigger và function:** Phát triển các trigger và function để tự động hóa và nâng cao hiệu suất của hệ thống.

 **Phát triển dự án:**

* **Chức năng đăng ký:** Viết mã phần backend để người dùng có thể đăng ký tài khoản mới.
* **Chức năng quản lý sinh viên và nhân viên:** Xây dựng giao diện và chức năng để admin có thể quản lý thông tin sinh viên và nhân viên.