# Ứng dụng quản lý trụ sạc xe điện

## 1. Chức năng hệ thống

* **Quản lí trụ sạc**
  + Thêm / sửa / xóa trạm (ID, tọa độ, loại sạc, số cổng).
  + Xem tình trạng trạm (tổng cổng, cổng trống/đang sạc/bảo trì).
* **Quản lý cổng sạc**
  + Theo dõi trạng thái cổng: trống / đang sạc / bảo trì / đặt trước.
  + Cập nhật khi **bắt đầu** hoặc **kết thúc** phiên sạc; ghi start/end/ETA (tùy chọn).
* **Đặt trước chỗ sạc**
  + Quy trình: chọn trạm → chọn cổng → chọn thời gian.
  + Hủy đặt chỗ; tự hủy khi quá **grace time**.
  + Kiểm tra trùng giờ, cảnh báo xung đột.
* **Quản lý khách hàng**
  + Thêm / sửa / xóa thông tin khách hàng.
  + Tra cứu khách hàng theo ID/SĐT.
  + Lưu lịch sử sạc và hóa đơn nội bộ.
* **Gợi ý trạm gần nhất (offline)**
  + Nhập vị trí hiện tại → đề xuất trạm gần nhất còn chỗ & đúng loại sạc.
* **Tối ưu lộ trình**
  + Tìm đường ngắn nhất đến trạm đích trong mạng lưới trạm–đường.
* **Hàng chờ tại trạm**
  + Khi hết cổng trống → quản lý danh sách chờ theo thứ tự đến (FIFO).
* **Báo cáo và thống kê**
  + Lượt sử dụng theo trạm/cổng, thời gian chờ trung bình.
  + Doanh thu ngày / tháng / năm; xuất CSV.
  + Top trạm bận nhất / giờ cao điểm.

## 2. Cấu trúc dữ liệu (CTDL)

* **Graph** (danh sách kề hoặc ma trận kề): mô hình mạng trạm & đường (trọng số = thời gian/quãng đường).
* **Hash table**: tra cứu trạm theo ID (O(1) trung bình).
* **vector<vector<bool>> / bitset**: trạng thái cổng của mỗi trạm (ma trận cổng).
* **map / set theo thời gian**: lịch đặt chỗ từng cổng (phát hiện trùng giờ).
* **queue**: hàng chờ tại trạm khi đầy cổng.
* **stack**: lịch sử thao tác (undo/redo nếu cần).
* **unordered\_map (Hash Table)**: khách hàng, tra cứu nhanh theo ID/SĐT.
* **BST (std::set / std::map)**: sắp xếp trạm theo lượt sử dụng/doanh thu.
* **priority\_queue**: Top-N trạm gần nhất hoặc điểm cao nhất khi gợi ý; heap ETA cổng sắp xong.

## 3. Thuật toán

* **Tính khoảng cách**: Euclid (tọa độ phẳng) / Haversine (lat/lon).
* **Gợi ý trạm gần nhất (offline)**: lọc trạm phù hợp loại sạc & còn cổng → tính khoảng cách → **sort** hoặc **min-heap** (Top-k).
* **Tối ưu lộ trình**: **Dijkstra** (trọng số thời gian/quãng đường) / **BFS** (ít chặng).
* **Đặt chỗ – kiểm tra trùng**: **set.lower\_bound** trên time slot mỗi cổng.
* **Sắp xếp & tra cứu**: **QuickSort / std::sort**; tìm kiếm **tuyến tính / nhị phân**.
* **Hàng chờ & hoàn tác**: **queue** (FIFO), **stack** (undo/redo).
* **Ước lượng hoàn tất (ETA – tùy chọn)**: ETA = start + (E\_req / P\_thực); quản lý cổng sắp xong bằng **min-heap (ETA)**.

## 4. Phân quyền (4 vai trò) & chức năng có thể khai thác

**A) Admin**

* **Quyền**: cao nhất; cấu hình hệ thống, tài khoản & phân quyền, sao lưu/khôi phục, import/export, xem toàn bộ audit log.
* **Chức năng khai thác**:
  + Tạo/sửa/xóa **trạm/cổng**; cấu hình **bảng giá**.
  + Bật/tắt tính năng: grace time, hàng chờ, ETA.
  + Quản lý người dùng hệ thống (tạo tài khoản Manager/Employee/User).

**B) Quản lý (Manager)**

* **Quyền**: vận hành chính.
* **Chức năng khai thác**:
  + CRUD **đặt chỗ**; chống trùng; quản lý **hàng chờ**.
  + **Start/Stop** phiên sạc; chỉnh **ETA**; in hóa đơn; xuất **báo cáo/CSV**.
  + Đặt **trạng thái cổng** (trống/đang sạc/đặt trước/bảo trì – khi có lịch).
  + Xem **audit log** (read-only). *(Có thể được Admin cấp quyền override trong ca khẩn, bắt buộc ghi log.)*

**C) Nhân viên (Employee/Operator)**

* **Quyền**: thao tác trong ca, không sửa hạ tầng/giá.
* **Chức năng khai thác**:
  + **Check-in / Start / Stop** phiên sạc; gán cổng; in phiếu.
  + Tạo/hủy **đặt chỗ** đơn giản; quản lý **queue** trong ca.
  + Xem tình trạng trạm/cổng, xem bảng giá, xem lịch trống.

**D) Người dùng (User/Kiosk – chỉ đọc)**

* **Quyền**: tra cứu tại chỗ (local only).
* **Chức năng khai thác**:
  + Xem **trạm/cổng trống**, khung giờ rảnh, **giá niêm yết**.
  + Nhập vị trí → nhận **gợi ý trạm gần nhất** (offline).
  + *Không có quyền ghi dữ liệu hoặc đặt chỗ trực tiếp.*

# PHẦN MỀM TÌM KIẾM GIA SƯ VÀ HỌC VIÊN

1. Các chức năng chính

### 1. Đăng ký và đăng nhập

* Người dùng (Tutor / Student) đăng ký tài khoản với thông tin cá nhân: tên, email, số điện thoại, môn học, khu vực, v.v.
* Đăng nhập bằng **username / password**, có phân quyền:
  + **Admin**: Quản lý tài khoản người dùng.
  + **Tutor**: Cập nhật hồ sơ, đăng lịch dạy.
  + **Student**: Đăng ký học, tìm gia sư.

### 2. Quản lý hồ sơ

* **Tutor**: Tạo / cập nhật hồ sơ (môn dạy, kinh nghiệm, giá tiền, khu vực, lịch rảnh).
* **Student**: Cập nhật thông tin cá nhân, nhu cầu học (môn, trình độ, thời gian).
* Phân quyền:
  + Admin + chính chủ tài khoản: chỉnh sửa hồ sơ.
  + Hồ sơ tutor: công khai cho mọi người xem.

### 3. Tìm kiếm gia sư / học viên

* **Student**: Tìm gia sư theo môn học, khu vực, giá, đánh giá.
* **Tutor**: Tìm học viên phù hợp chuyên môn.
* Phân quyền:
  + Student: Xem danh sách gia sư, gửi yêu cầu học.
  + Tutor: Xem yêu cầu học từ học viên, chấp nhận / từ chối.

### 4. Quản lý lịch học

* **Tutor**: Đăng lịch dạy (ngày, giờ, địa điểm/online).
* **Student**: Đặt lịch học với gia sư, xem lịch đã đặt.
* Phân quyền:
  + Admin: Xem tất cả lịch học, giải quyết tranh chấp.
  + Tutor: Chỉnh sửa lịch dạy của mình.
  + Student: Xem / xác nhận lịch học liên quan.

### 5. Đánh giá và phản hồi

* **Student**: Đánh giá gia sư sau buổi học (1–5 sao, bình luận).
* **Tutor**: Có thể đánh giá học viên.
* Phân quyền:
  + Chỉ student đã học mới được đánh giá tutor.
  + Admin: Xem / xóa đánh giá không phù hợp.

### 6. Quản lý thanh toán

* **Student**: Thanh toán học phí (giả lập hoặc lưu lịch sử giao dịch).
* **Tutor**: Xem lịch sử nhận thanh toán.
* Phân quyền:
  + Admin: Quản lý giao dịch, xử lý khiếu nại.
  + Tutor / Student: Chỉ xem giao dịch của mình.

### 7. Quản lý hệ thống

* **Admin**:
  + Thêm / xóa tài khoản.
  + Duyệt hồ sơ gia sư.
  + Xử lý báo cáo vi phạm.

B. CTDL & TT

### 1. Tìm kiếm

* **Linear Search**: Với dữ liệu ít hoặc chưa sắp xếp.
* **Binary Search**: Khi danh sách đã được sắp xếp (VD: tìm gia sư theo học phí).

### 2. Sắp xếp

* **Quick Sort / Merge Sort**: Sắp xếp danh sách gia sư theo:
  + Học phí.
  + Kinh nghiệm.
  + Độ phù hợp với nhu cầu học viên.

### 3. Cấu trúc cây

* **Binary Search Tree (BST)**: Lưu trữ / tra cứu hồ sơ theo ID, tên, môn học.
* **AVL Tree (Cân bằng)**: Tăng tốc tìm kiếm khi dữ liệu lớn.

### 4. Hashing

* **Hash Table**: Quản lý tài khoản nhanh, tìm kiếm theo tên/email.
* Lưu danh sách tutor/student để thao tác tìm kiếm & sắp xếp.

### 5. Mảng / Vector

* Lưu danh sách gia sư hoặc học viên → thao tác tìm kiếm, sắp xếp.

### 6. Danh sách liên kết (Linked List)

* Quản lý các danh sách động: lịch học, đánh giá, giao dịch.