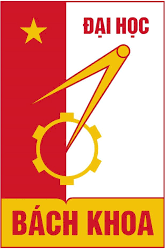
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



**Software design description**

**Đồ án các công nghệ xây dựng hệ thống thông tin - IT4421**

**Đề tài: Hệ thống thu phí tại các điểm đỗ xe công cộng - Parking Fee**

**Giáo viên hướng dẫn: TS. Đào Thành Chung**

|  |  |
| --- | --- |
| Trần Văn Hiếu | 20151369 |
| Phan Tuấn Anh | 20150157 |
| Tạ Thị Minh Huyền | 20151749 |
| Đào Văn Nghĩa | 20152644 |

**Mã tài liệu: SDD-v2.1**

**Hà Nội, Tháng 10 năm 2018**

# **CHƯƠNG 1: GIỚI THIỆU**

**1.1. Mục đích**

* Tài liệu này là bản thiết kế chi tiết hệ thống, bao gồm các thiết kế: phân tích chi tiết lớp (analyst class), cơ sở dữ liệu (database), giao diện người dùng (graphic userinterface).
* Dựa trên phân tích có trong tài liệu đặc tả SRS đưa ra quy trình hoạt động của hệ thống thông qua các biểu đồ phân tích chi tiết lớp và biểu đồ tuần tự (sequence diagram) tương ứng cho các hành vi của từng ca sử dụng.
* Đưa ra thiết kế cơ sở dữ liệu cụ thể, cơ sở dữ liệu cho phép lưu trữ thông tin về người dùng, thông tin các vị trí đỗ xe trong bãi đỗ xe, đồng thời lưu vết lịch sử đỗ xe tại các điểm thuận tiện cho việc thống kê của người quản lý.
* Từ tài liệu thiết kế này, nhóm phát triển hệ thống có cơ sở để lập trình phát triển, hình thành hệ thống cho người dùng. Thuận tiện cho việc kết nối các module của hệ thống với nhau.

**1.2. Cấu trúc tài liệu**

Bản thiết kế này chia làm 5 phần chính:

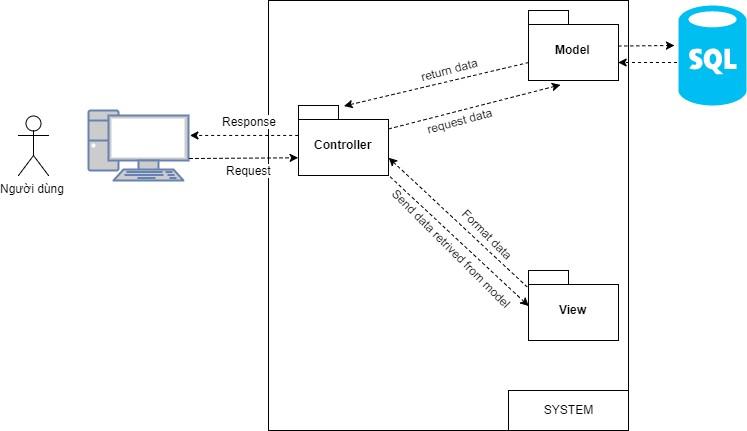
* 1. Mô hình hóa cấu trúc hệ thống: Cấu trúc hệ thống tổng quan
  2. Thiết kế lớp chi tiết: các lớp tham gia hệ thống
  3. Mô hình hóa hành vi hệ thống: Trình tự hoạt động
  4. Thiết kế cơ sở dữ liệu
  5. Thiết kế giao diện người dùng

**CHƯƠNG 2: MÔ HÌNH HÓA CẤU TRÚC HỆ THỐNG**

**2.1. Các tầng và mối quan hệ trên hệ thống**

2.1.1. Hệ thống được xây dựng dựa theo mô hình MVC

* Tổng quan mô hình MVC (Model – View - Controller)



*Hình 1. Mô hình MVC*

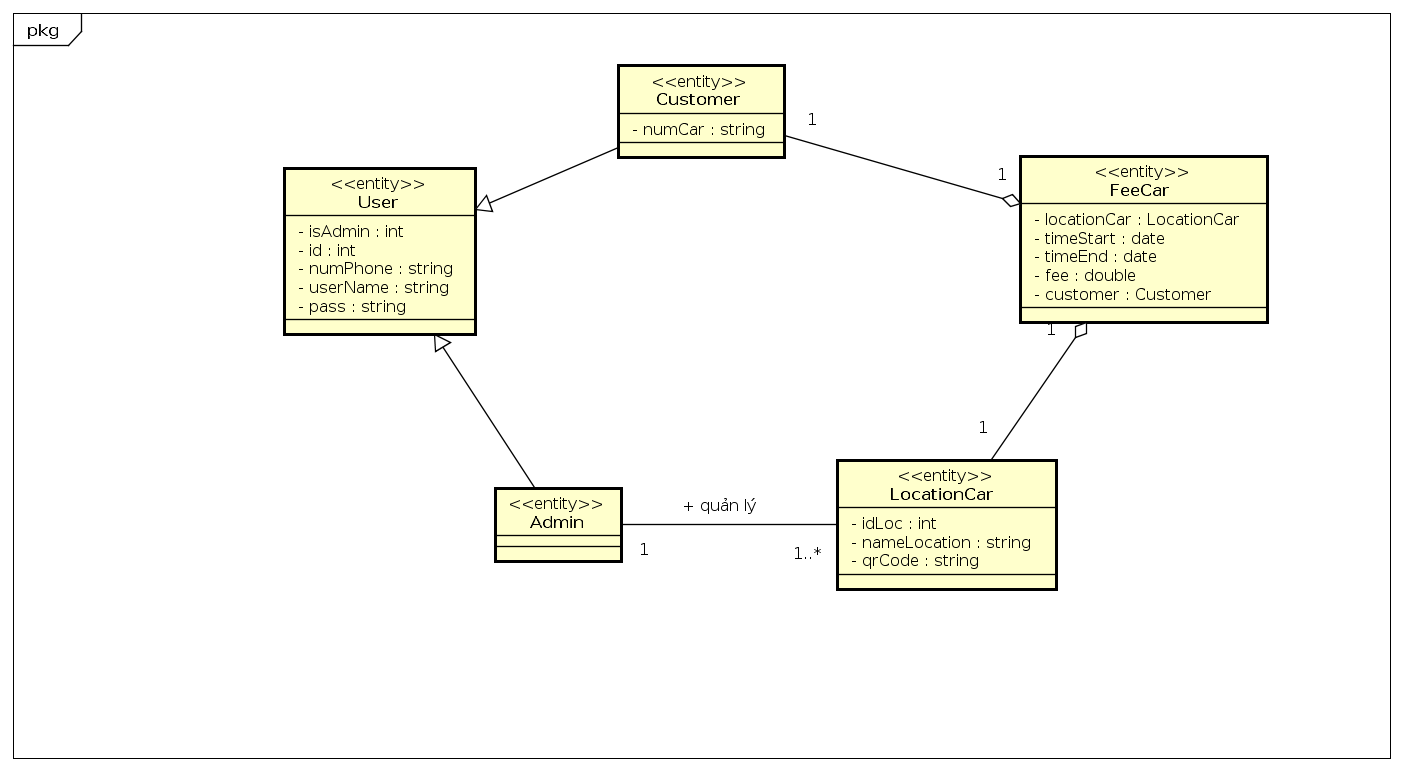
* + Chức năng của từng tầng trong mô hình MVC
    - **Controller:** Đây là nơi đầu tiền nhận các yêu cầu (requests), phân tích yêu cầu, khởi tạo và gọi model, sau đó nhận các hồi đáp (response) từ model và gửi ra các lớp giao diện (view). Trong thực tế, Controller được gọi từ điểm đầu vào của ứng dụng(mà thường là các file index.php hoặc index.html). Tập tin này sẽ giao toàn bộ các yêu cầu gửi từ client cho controller xử lý.
    - **Model**: Đại diện cho dữ liệu và logic của ứng dụng, thường hay gọi là business logic, có trách nhiệm:
* Thêm, sửa, xoá dữ liệu…
* Là nơi thực thi logic nghiệp vụ của ứng dụng.
  + - **View**: Chịu trách nhiệm định đạng lại dữ liệu được truyền ra từ model. Dữ liệu được truyền ra có thể có nhiều định dạng khác nhau tùy vào cách người lập trình xử lý như là xml, json, array.

2.1.2. **Mô tả kịch bản hệ thống hướng theo mô hình MVC**

* Kịch bản đăng nhập: User gửi yêu cầu đăng nhập đến Controller(LogInController), LogInController tiếp nhận request, LoginController lấy kết quả xác định tồn tại user hay không từ lớp Model(User), rồi gọi lớp View tương ứng với các trường hợp tồn tại và không tồn tại user trong database.
* Kịch bản đăng kí: User gửi yêu cầu đăng kí tài khoản mới đến Controller( SingUpController), SingUpController tiếp nhận request, SingUpController lấy kết quả xác định tồn tại user hay không từ lớp Model(User), nếu chưa tồn tại, cho phép đăng kí, nếu tồn tại không cho phép đăng kí, sau đó, controller này gọi đến 1 lớp view trong package View.
* Kịch bản Xử lý QRCode: User gửi yêu cầu xác nhận mã QRCode đếnController(ProcessQRcodeController), ProcessQRcodeController tiếp nhận request, ProcessQRcodeController lưu lại thông tin QRcode cùng tài khoản của user trong database, sau đó, controller này gọi đến 1 lớp view trong package View, và hiển thị thông tin đã lưu thành công mã QRCode cho người dùng.
* Kịch bản Xử lí thông tin từ Sensor: Sensor gửi yêu cầu lưu lại thờigian đến Controller(SaveTimeStartController và SaveTimeEndController), 2 controller này tiếp nhận request, rồi lưu lại thông tin thời gian mà xe đó dừng và đỗ tại bãi xe cùng tài khoản của user trong database, sau đó, 2 controller này gọi đến 1 lớp view trong package View, và hiển thị thông tin đã lưu thành công thời gian đỗ xe cho người dùng.
* Kịch bản tính tiền gửi xe cho khách hàng: Khi người dùng kết thúcquá trình gửi xe, sensor gửi tín hiệu (request) cho controller (SaveTimeEndController), controller này save thông tin thời gian cuối cùng vào database, sau khi lưu thông tin thành công, controller này gọi đến ComputeFeeController, ComputeFeeController lấy thông tin từ Model(FeeCar), lấy được thông tin đầu và cuối của user, rồi controller lấy tiếp thông tin bảng giá từ Model(LevelFee), tính toàn tiền phải trả, tiếp đó, nó gọi đến 1 class View trong package View, hiển thị thông tin số tiền phải trả cho khách hàng.
* Kịch bản thanh toán tiền gửi xe cho khách hàng: Sau khi ngườidùng gửi yêu cầu thanh toán đến PayMentController, PayMentController dùng thông tin tài khoản user, lấy thông tin số dư tài khoản từ tải khoản đã có từ lớp Model(Account), trừ đi số tiền phải trả sau đó lưu thông tin số tiền còn lại vào database, tiếp đó controller này gọi đến 1 lớp view trong Package View nhằm hiển thị cho người trạng thái của việc thành công(đã thanh toán thành công, số tiền của người dùng không đủ trong quá trình thanh toán, v.v ..).

2.2. Sơ đồ triển khai hệ thống

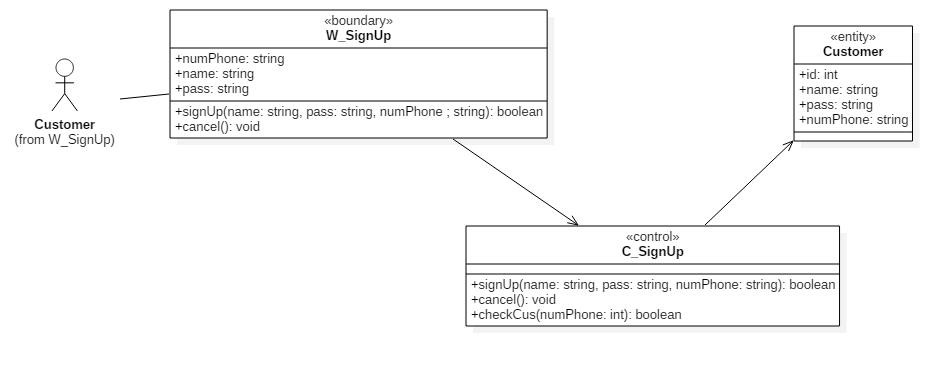
2.3. Mô hình hoá lĩnh vực ứng dụng



2.4. Các lớp tham gia ca sử dụng.

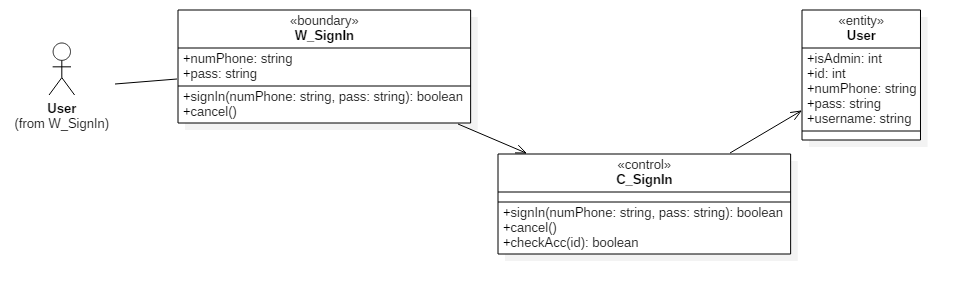
Các lớp tham gia ca sử dụng gồm 3 loại:  
❖ Boundaries (các lớp biên hay các lớp đối thoại): tương ứng với tầng View  
❖ Entities (các lớp thực thể hay các lớp lĩnh vực): tương ứng với tầng Model  
❖ Controls (các lớp điều khiển): tương ứng với tầng Controller

2.4.1. Ca sử dụng Đăng kí:



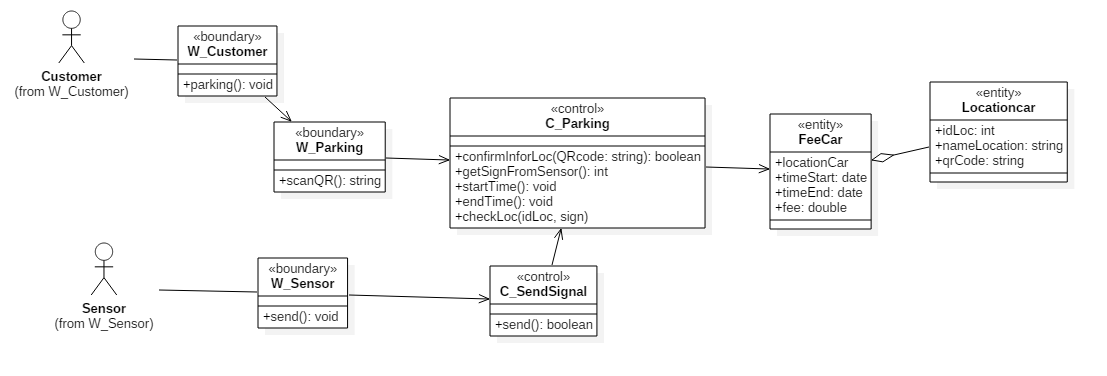
*Hình 2: Biểu đồ lớp tham gia ca sử dụng Đăng kí*

2.4.2. Ca sử dụng Đăng nhập



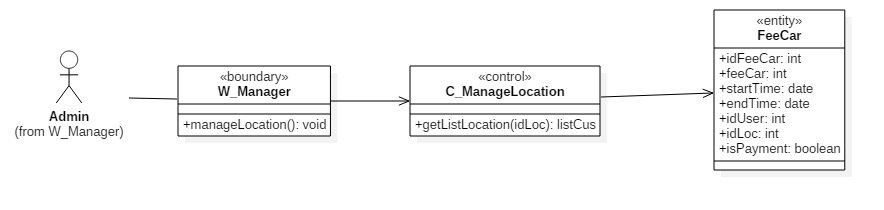
*Hình 3: Biểu đồ lớp tham gia ca sử dụng Đăng nhập*

2.4.3. Ca sử dụng Đăng kí đỗ xe của người dùng và Gửi thông tin đỗ xe của cảm biến



*Hình 4: Biểu đồ lớp tham gia ca sử dụng Đăng kí đỗ xe*

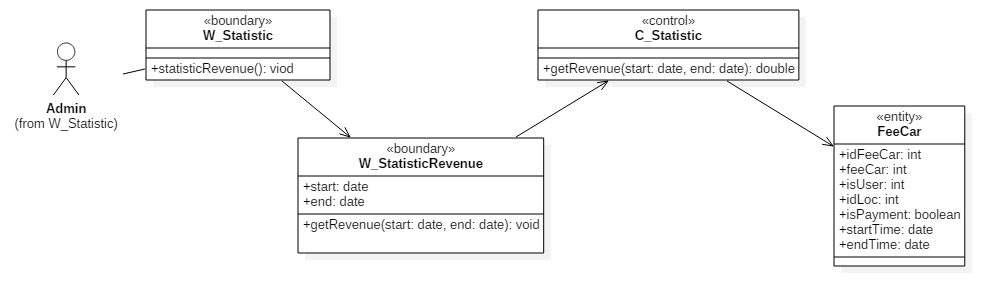
2.4.4. Ca sử dụng Quản lý thông tin bãi đỗ



*Hình 5: Biểu đồ lớp tham gia ca sử dụng Quản lý thông tin bãi đỗ*

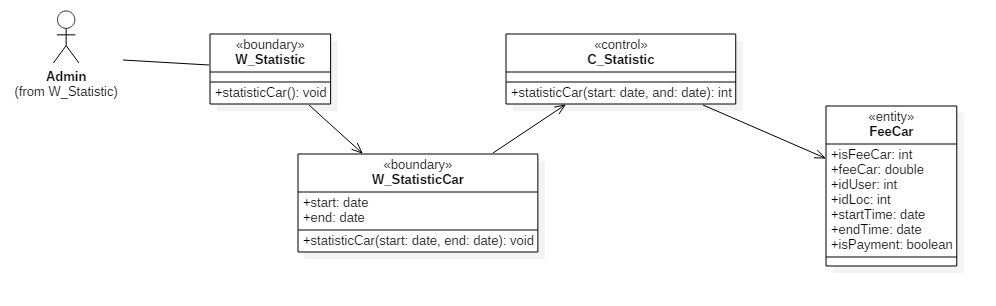
2.4.5. Ca sử dụng Thống kê bãi đỗ

2.4.5.1. Thống kê theo doanh thu



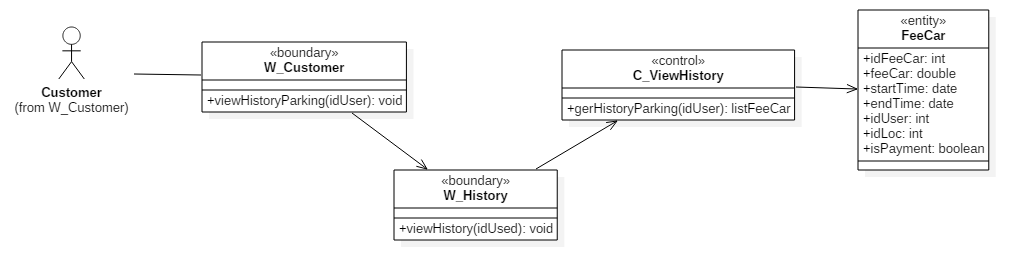
*Hình 6: Biểu đồ lớp tham gia ca sử dụng Thống kê theo doanh thu*

2.4.5.1. Thống kê theo lượng xe đỗ



*Hình 7: Biểu đồ lớp tham gia ca sử dụng Thống kê theo lượng xe đỗ*

2.4.6. Ca sử dụng Xem lịch sử đỗ xe



*Hình 8: Biểu đồ lớp tham gia ca sử dụng Xem lịch sử đỗ xe*

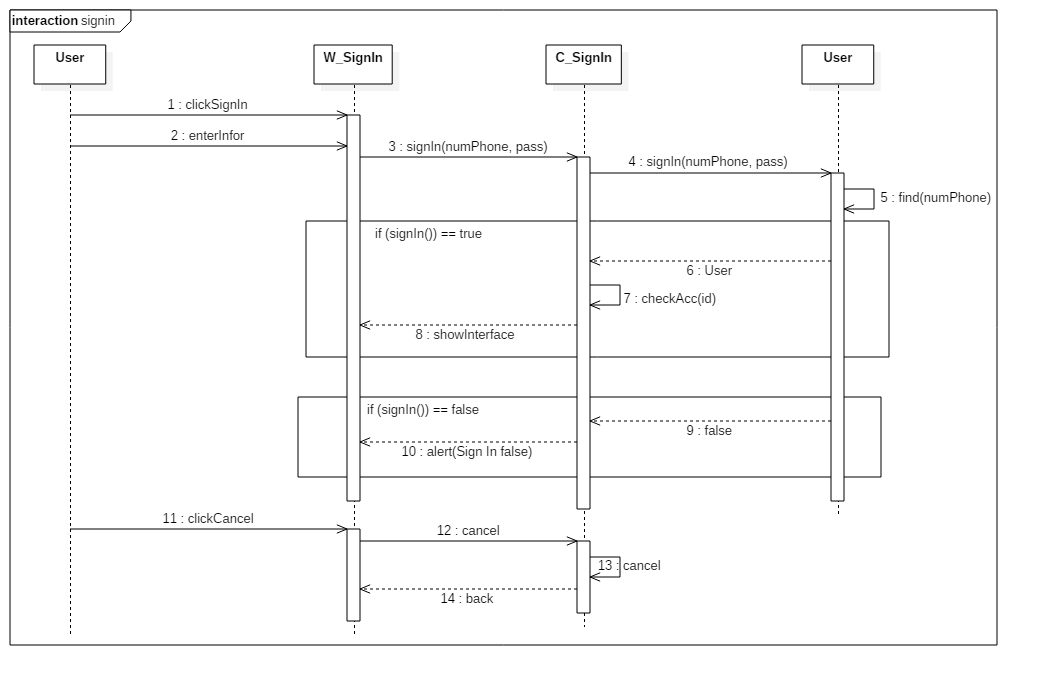
**CHƯƠNG 3: MÔ HÌNH HOÁ HÀNH VI HỆ THỐNG**

**3.1. Hành vi Đăng kí**

****

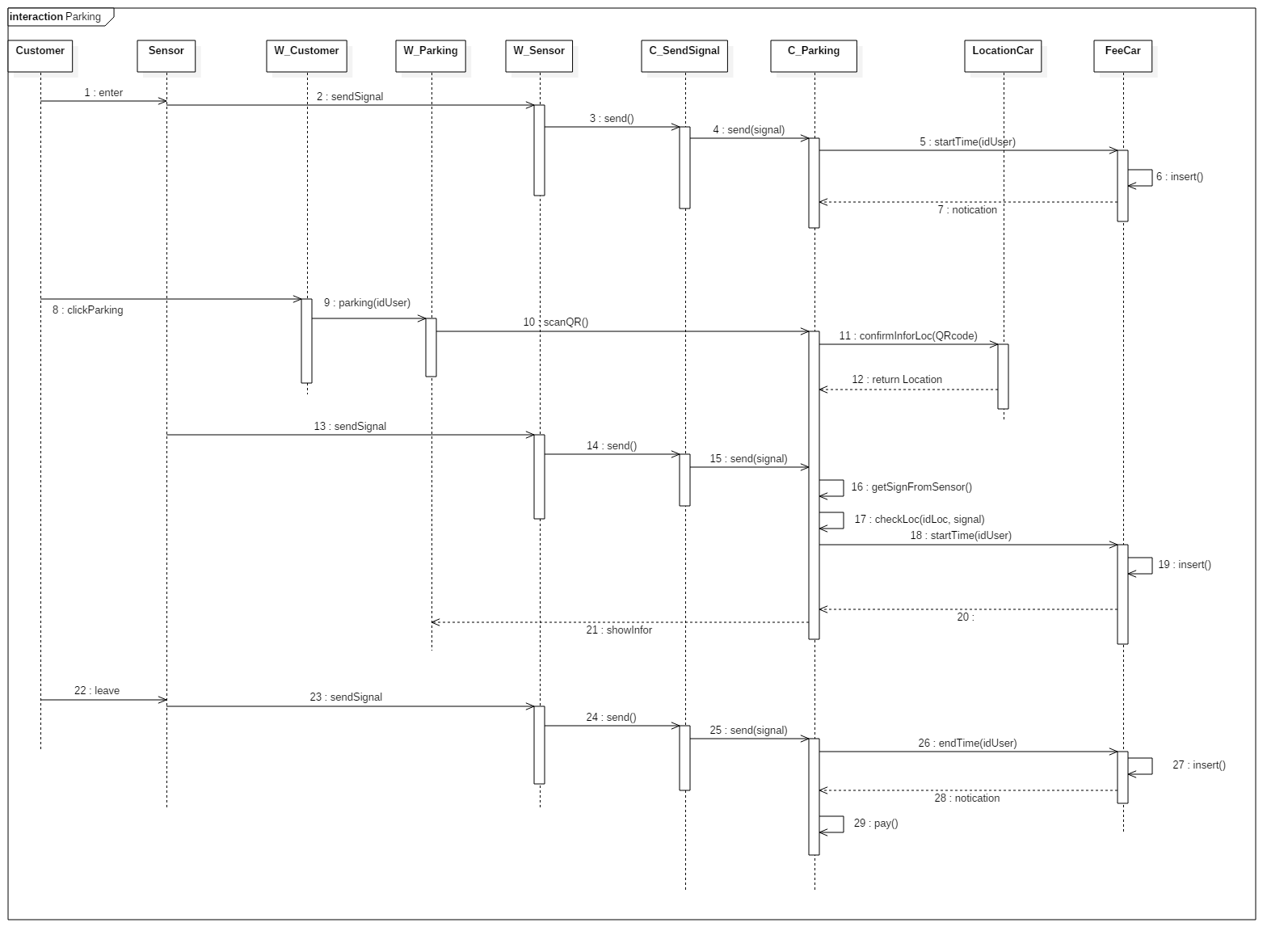
*Hình 9: Biểu đồ Hành vi Đăng kí*

**3.2. Hành vi Đăng nhập**

****

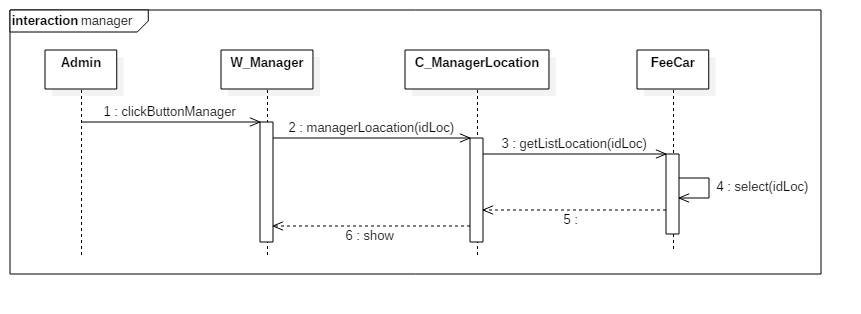
*Hình 10: Biểu đồ Hành vi Đăng nhập*

**3.3. Hành vi Đăng kí đỗ xe**

****

*Hình 11: Biểu đồ Hành vi Đăng kí đỗ xe*

**3.4. Hành vi Quản lý thông tin bãi đỗ**

****

*Hình 12: Biểu đồ Hành vi Quản lý thông tin bãi đỗ*

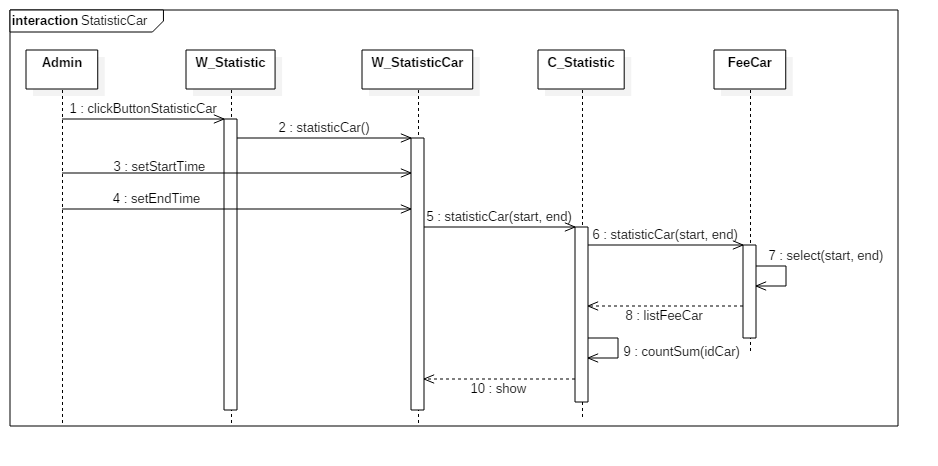
**3.5. Hành vi Thống kê**

3.5.1. Hành vi Thống kê theo doanh thu



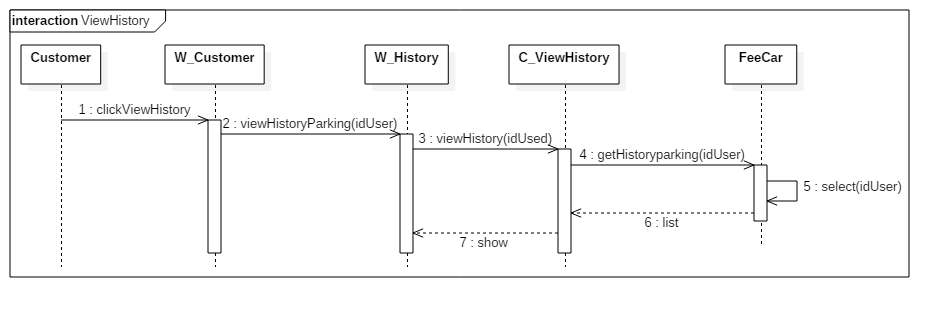
*Hình 13: Biểu đồ Hành vi Thống kê theo doanh thu*

3.5.1. Hành vi Thống kê theo số lượng xe đỗ



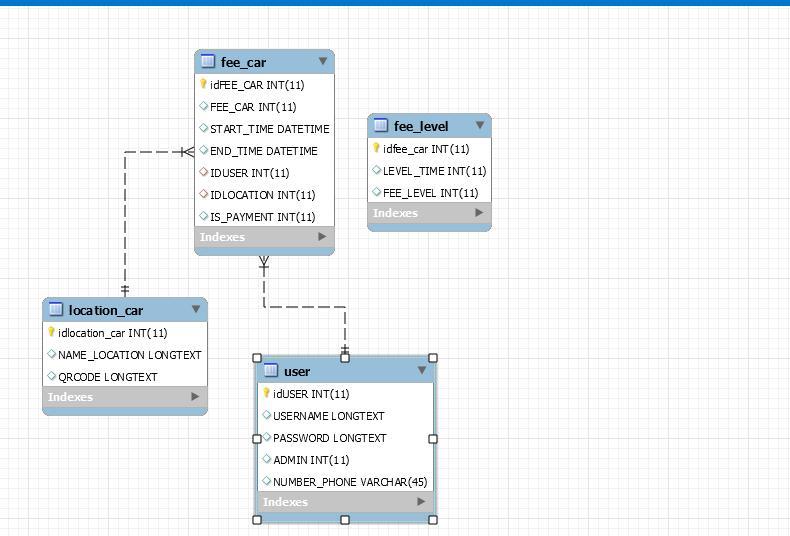
*Hình 14: Biểu đồ Hành vi Thống kê theo số lượng xe*

**3.6. Hành vi Xem lịch sử đỗ xe**

****

*Hình 15: Biểu đồ Hành vi Xem lịch sử đỗ xe*

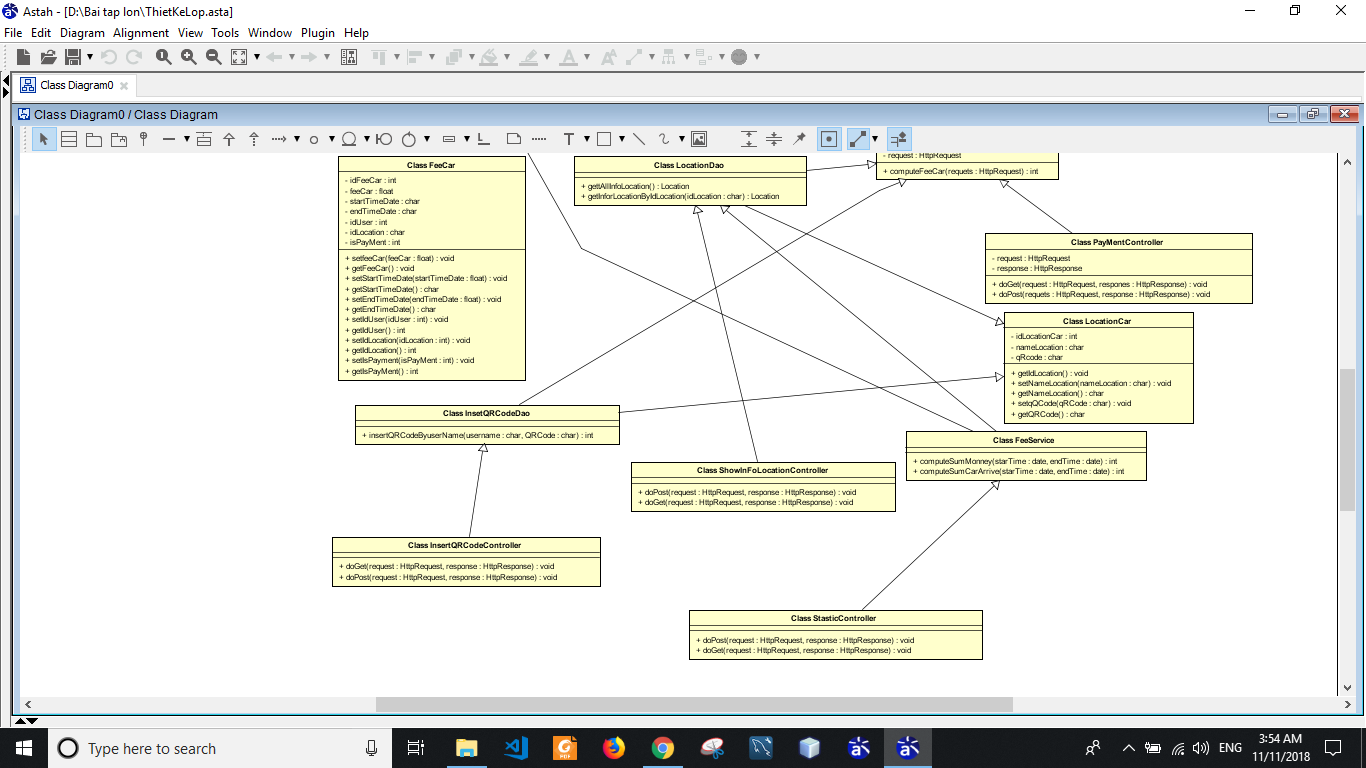
**CHƯƠNG 4: THIẾT KẾ CƠ SỞ DỮ LIỆU**

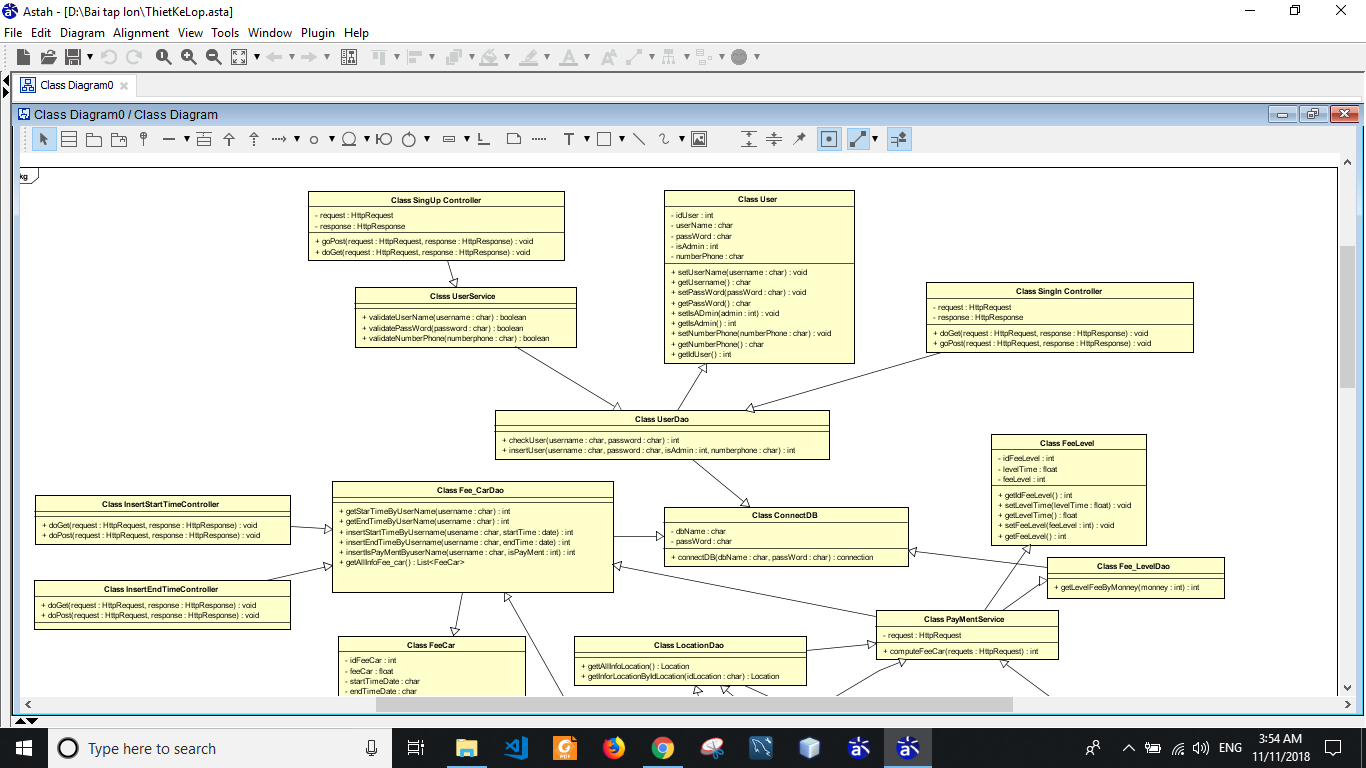


**A screenshot of a computer

Description generated with very high confidence**

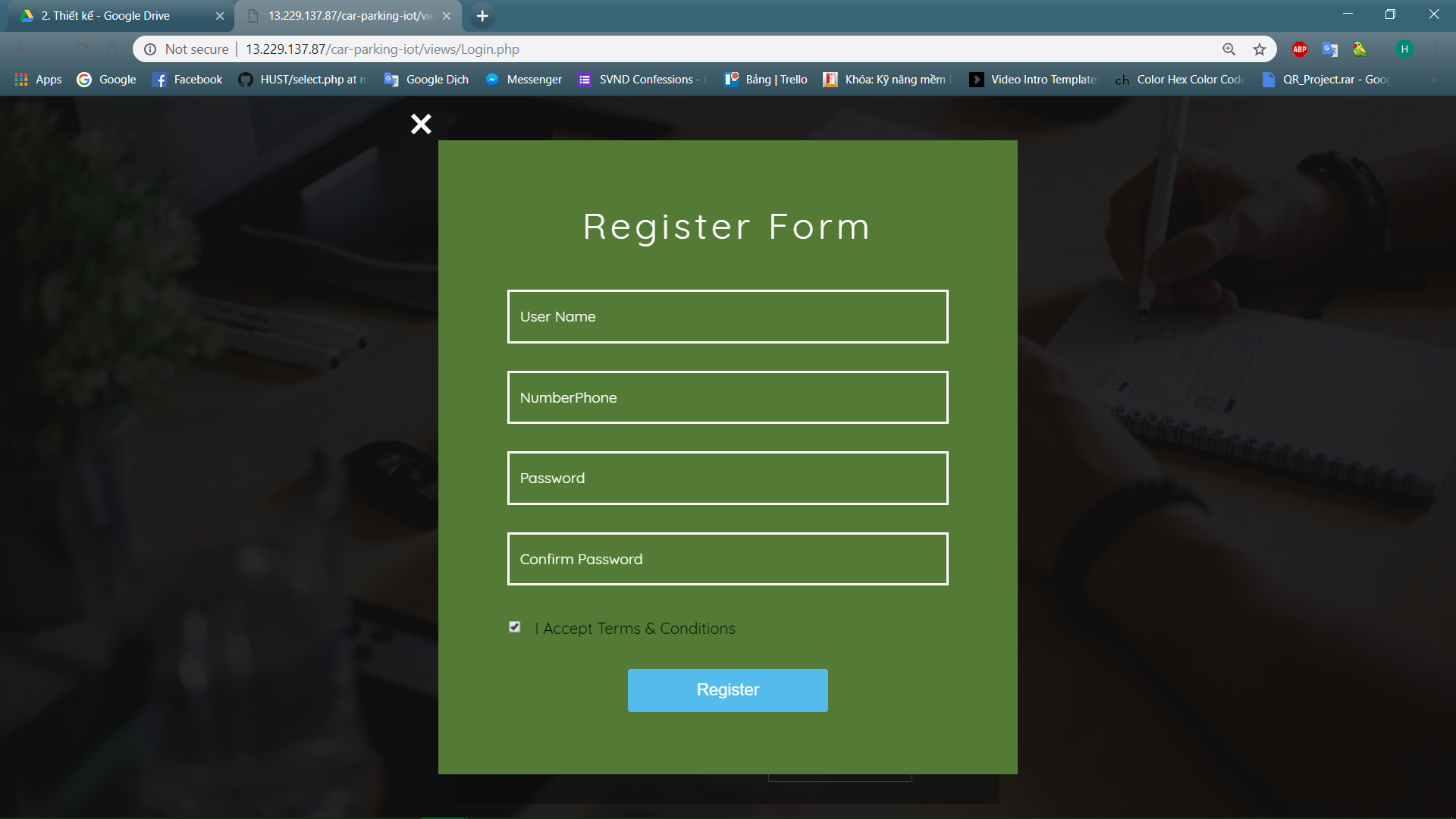
**CHƯƠNG 5: THIẾT KẾ LỚP CHI TIẾT**

****

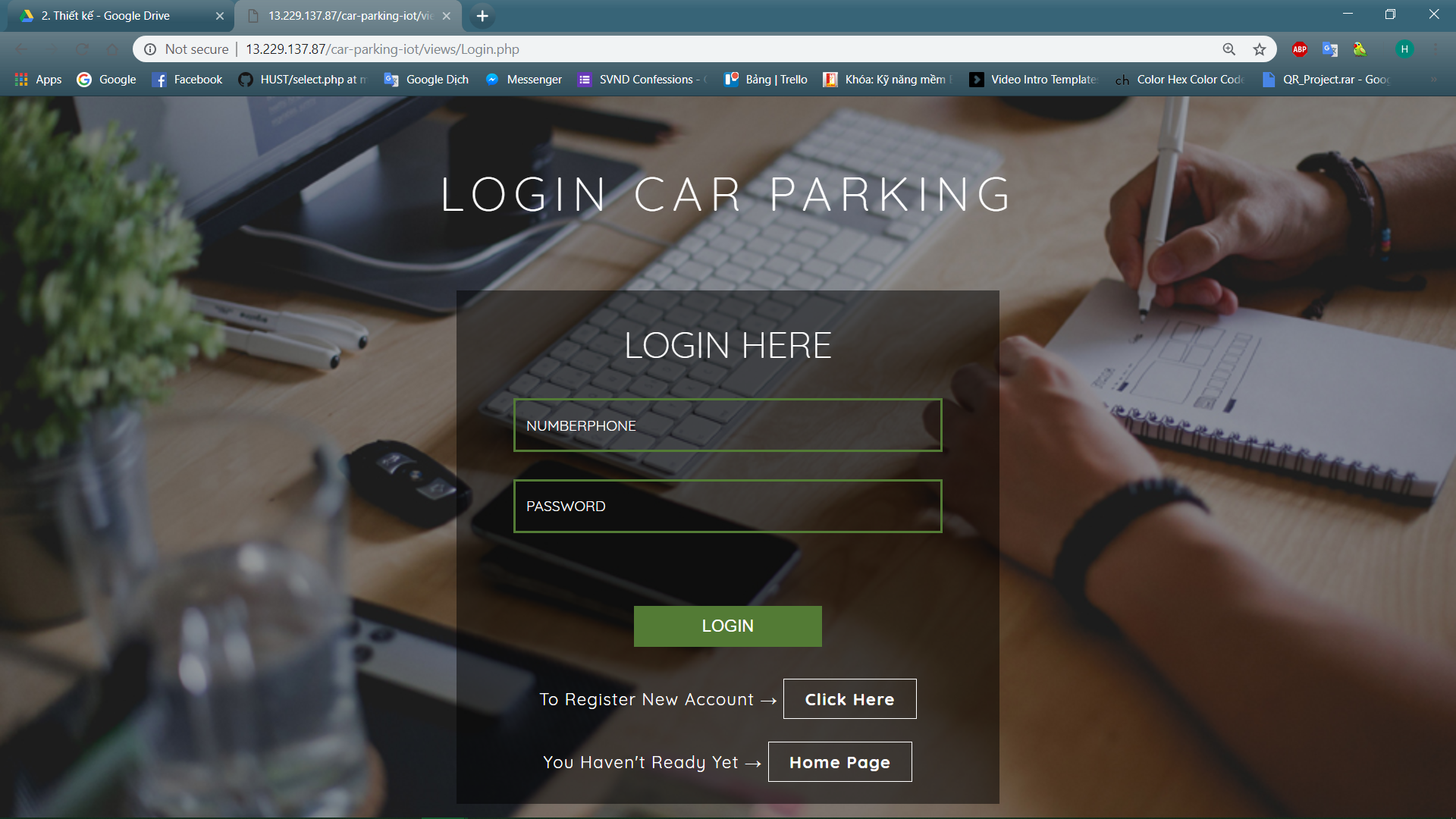
****

**CHƯƠNG 6: THIẾT KẾ GIAO DIỆN**

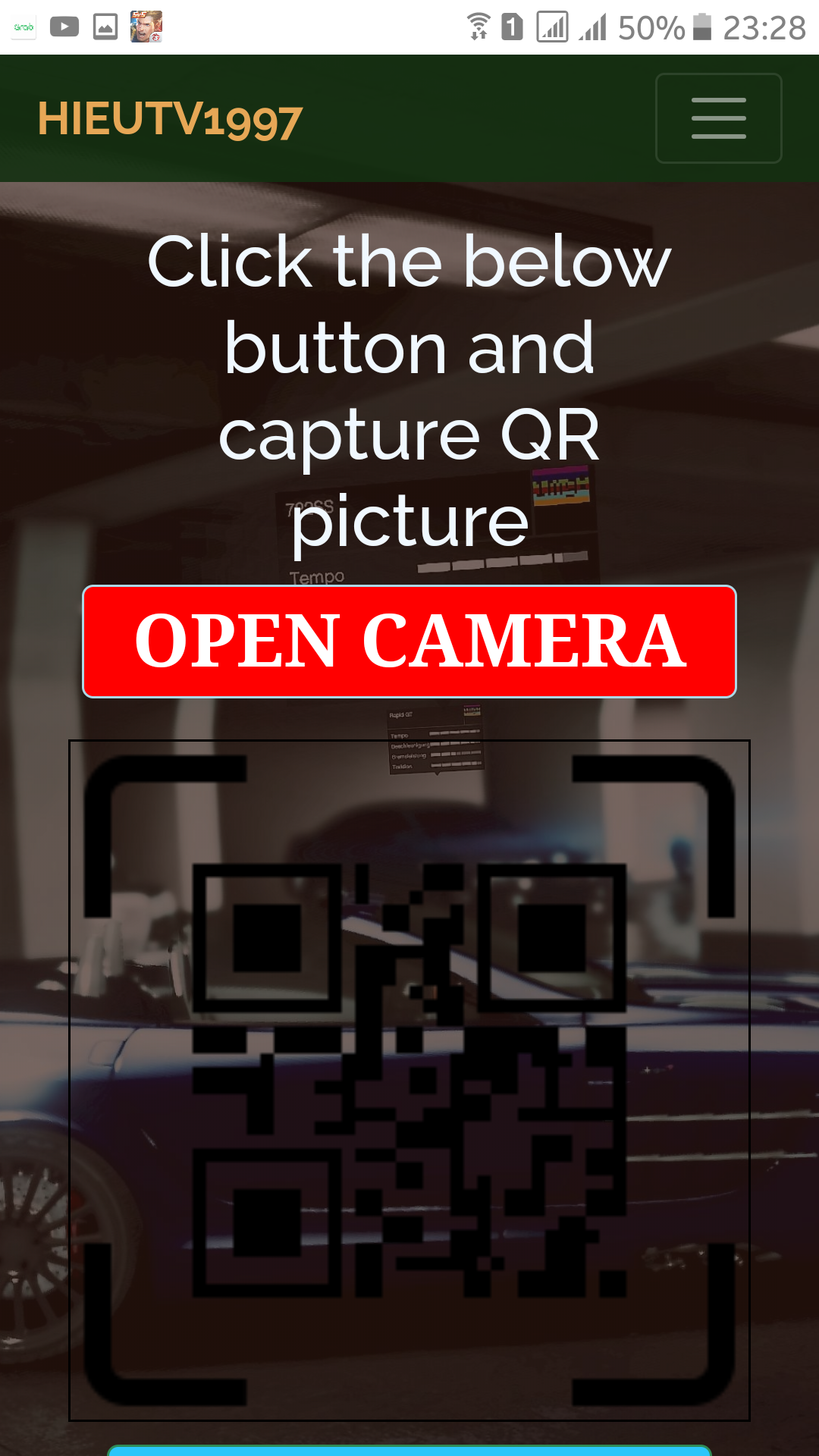
**5.1. Giao diện đăng kí**



**5.2. Giao diện đăng nhập**



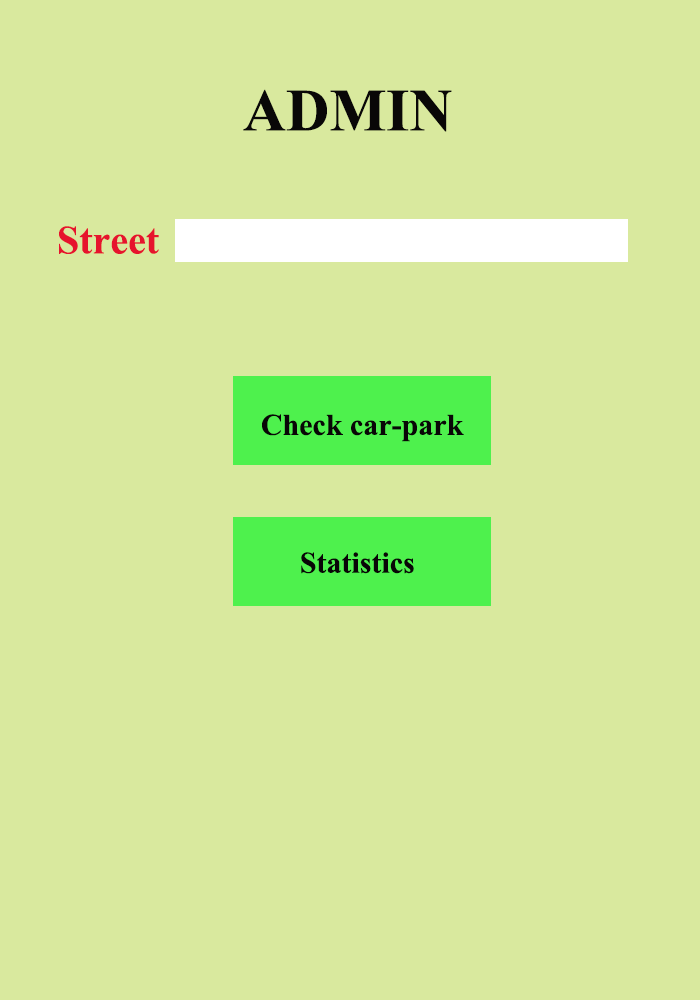
**5.3. Giao diện quét QR code**



**5.4. Giao diện thông báo đỗ xe thành công**

****

**5.5. Giao diện quản lí**

****