🎓 逢甲大學資工系學士後專班畢業專題報告

大型家具回收線上申請平台之數位化與管理架構實踐

Digital Transformation of Furniture Recycling: System Architecture and Management Implementation

---

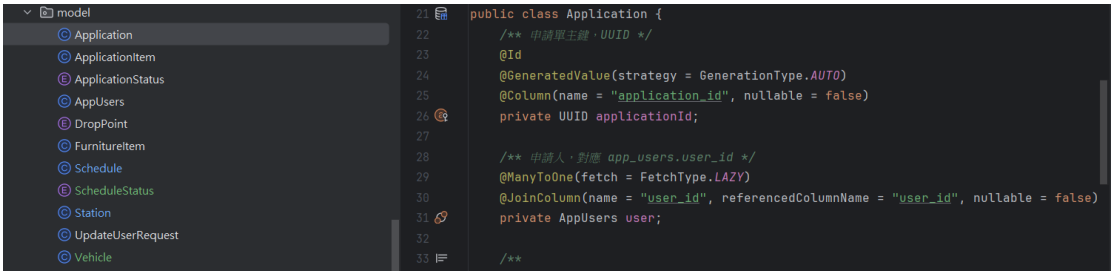
■ 第一部分：報告者資訊與專案背景

- **報告者姓名**：周彥廷
- **專案角色**：**後台管理系統主開發者 (Lead Admin System Developer)**
- **技術棧**：Java 21, Spring Boot 3.5, Hibernate 6, PostgreSQL (Supabase), Vue 3.
- **開發環境定位**：本系統設計之初即定位為**內部演示原型 (Prototype)**。為確保展示穩定性，主要運行於**本地端 (Localhost)** 環境進行技術驗證，並未規劃部署於公開生產環境。

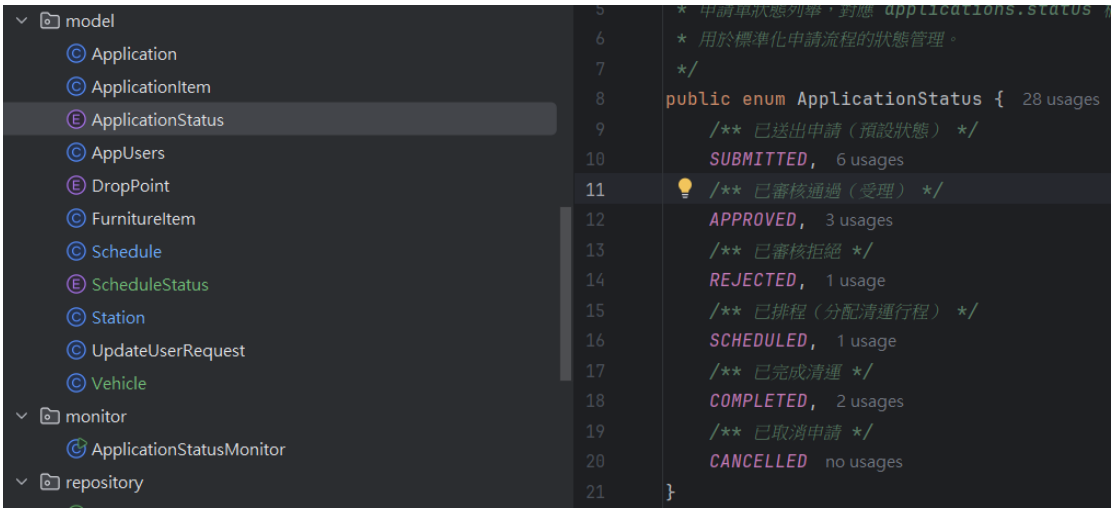
■ 第二部分：後台管理系統架構設計

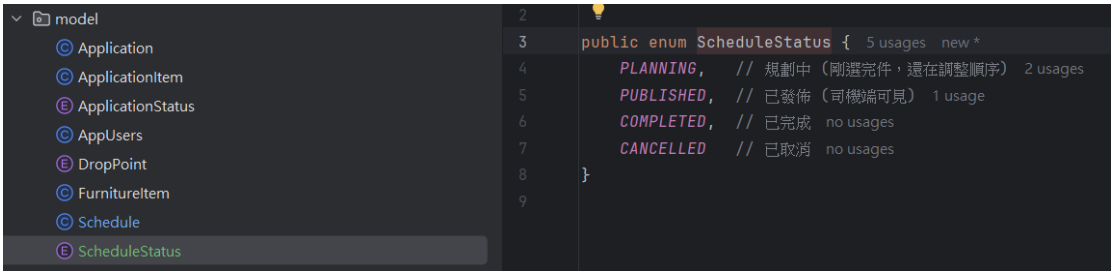
1. 實體建模與數據完整性 (Data Integrity)

- **UUID 主鍵策略**：全面採用 UUID 替代遞增整數，防止外部探測數據量，強化系統安全性。



- **狀態機設計**：實作 ApplicationStatus 與 ScheduleStatus 列舉（Enum），精確管控案件生命週期。





- **防禦性編程**：在 Application 實體層級實作 ensureEditable()，鎖定已受理案件，防止非法竄改。

2. **原子性排程調度引擎 (Atomic Scheduling)**

本人開發了具備高度一致性的「派車排程 API (/select-cases)」：

- **技術亮點**：使用 @Transactional 確保「建立 Schedule 紀錄」與「批量更新案件狀態」同步成功。
- **事務一致性**：若更新過程中發生任何異常，系統將自動回滾 (Rollback)，確保數據一致性。

```
@PostMapping("/select-cases")  new *
@Transactional
public ResponseEntity<?> selectCasesForSchedule(@RequestBody ScheduleSelectRequest requ
    try {
        // 1. 建立 Schedule 並填滿「所有」必填欄位
        Schedule schedule = new Schedule();
        schedule.setPlateNumber(request.getPlateNumber());
        schedule.setStatus(ScheduleStatus.PLANNING);

        // 🚨 修正：必須給予日期和時間，否則 SQL 會報 Constraint Violation
        schedule.setScheduleDate(LocalDate.now()); // 預設為今天
        schedule.setEta(java.time.Instant.now());    // 預設為現在

        // 2. 儲存 Schedule 取得真實 ID
        schedule = scheduleRepository.save(schedule);
        System.out.println("✅ 已建立排程，ID: " + schedule.getScheduleId());

        // 3. 更新關聯案件
        if (request.getCases() != null) {
            for (ScheduleSelectRequest.CaseSelection selection : request.getCases()) {
                Application app = applicationRepository.findById(selection.getApplicat
                        .orElseThrow(() -> new EntityNotFoundException("找不到案件: " +

                app.setSchedule(schedule); // 建立真關聯
                // 🚨 關鍵修正：將案件狀態改為 SCHEDULED (已排程)
                // 這樣它才會從「已通過/待處理」移動到「派車中」
                app.setStatus(ApplicationStatus.SCHEDULED);
                applicationRepository.save(app);
            }
        }
```

■ 第三部分：硬核技術實作：ORM 與資料流分析

1. 關聯載入策略優化

- **延遲載入 (Lazy Loading)**：為優化系統效能，對 User、Schedule 與 Station 的關聯均設定為 FetchType.LAZY，避免無謂的數據加載。

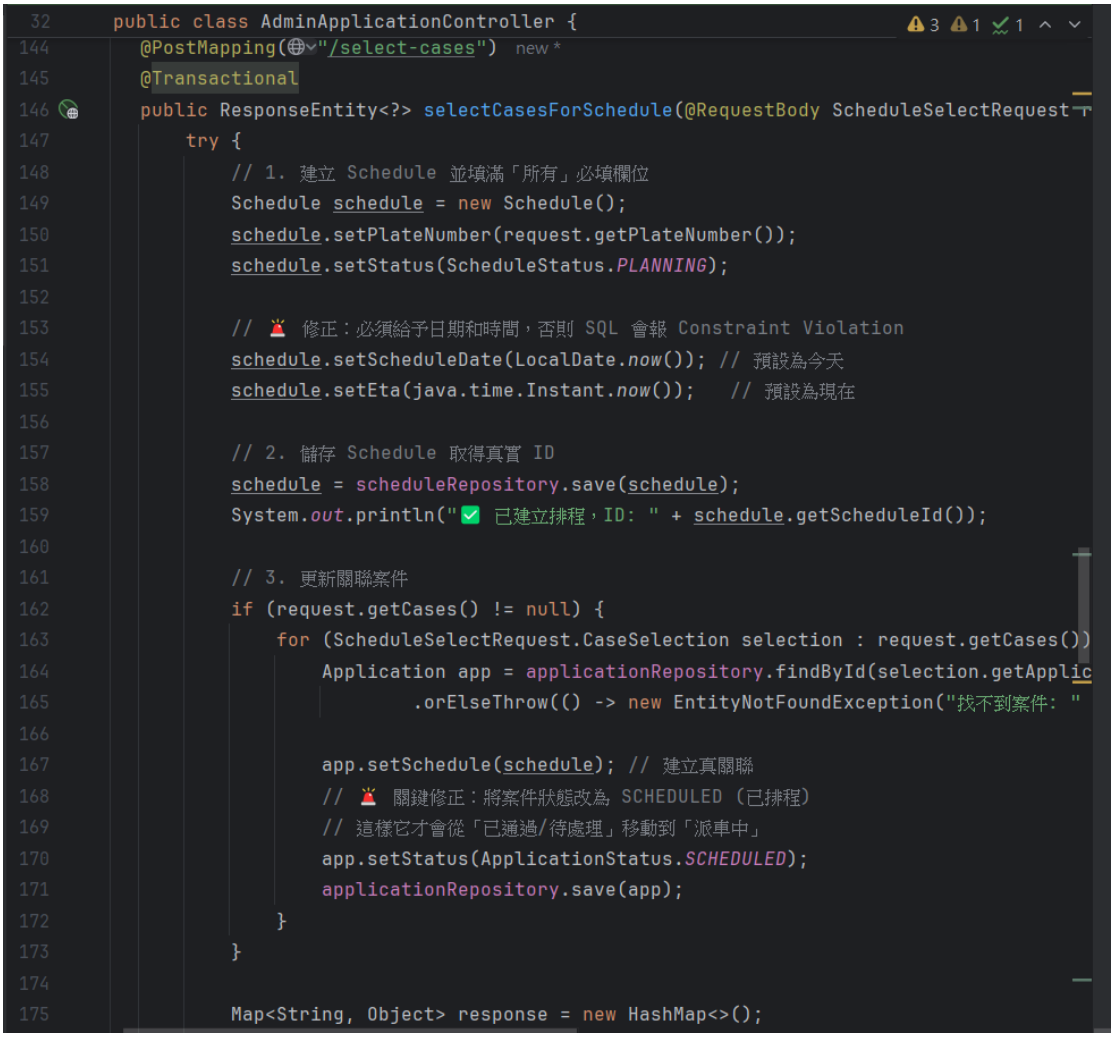
```
21    public class Application {                                                    9
28        /** 申請人，對應 app_users.user_id */
29        @ManyToOne(fetch = FetchType.LAZY)
30        @JoinColumn(name = "user_id", referencedColumnName = "user_id", nullable = false)
31        private AppUsers user;
32
33        /**
34         * 關聯：行程（Schedule），外鍵 schedule_id
35         * 單向關聯，Schedule 端無回指集合
36         */
37        @ManyToOne(fetch = FetchType.LAZY)
38        @JoinColumn(name = "schedule_id", nullable = true)
39        private Schedule schedule;
40
41        /**
42         * 關聯：清運站（Station），外鍵 station_id
43         * 對應 stations 資料表主鍵
44         */
45        @ManyToOne(fetch = FetchType.LAZY)
46        @JoinColumn(name = "station_id", referencedColumnName = "station_id", nullable = t
47        private Station station;
```

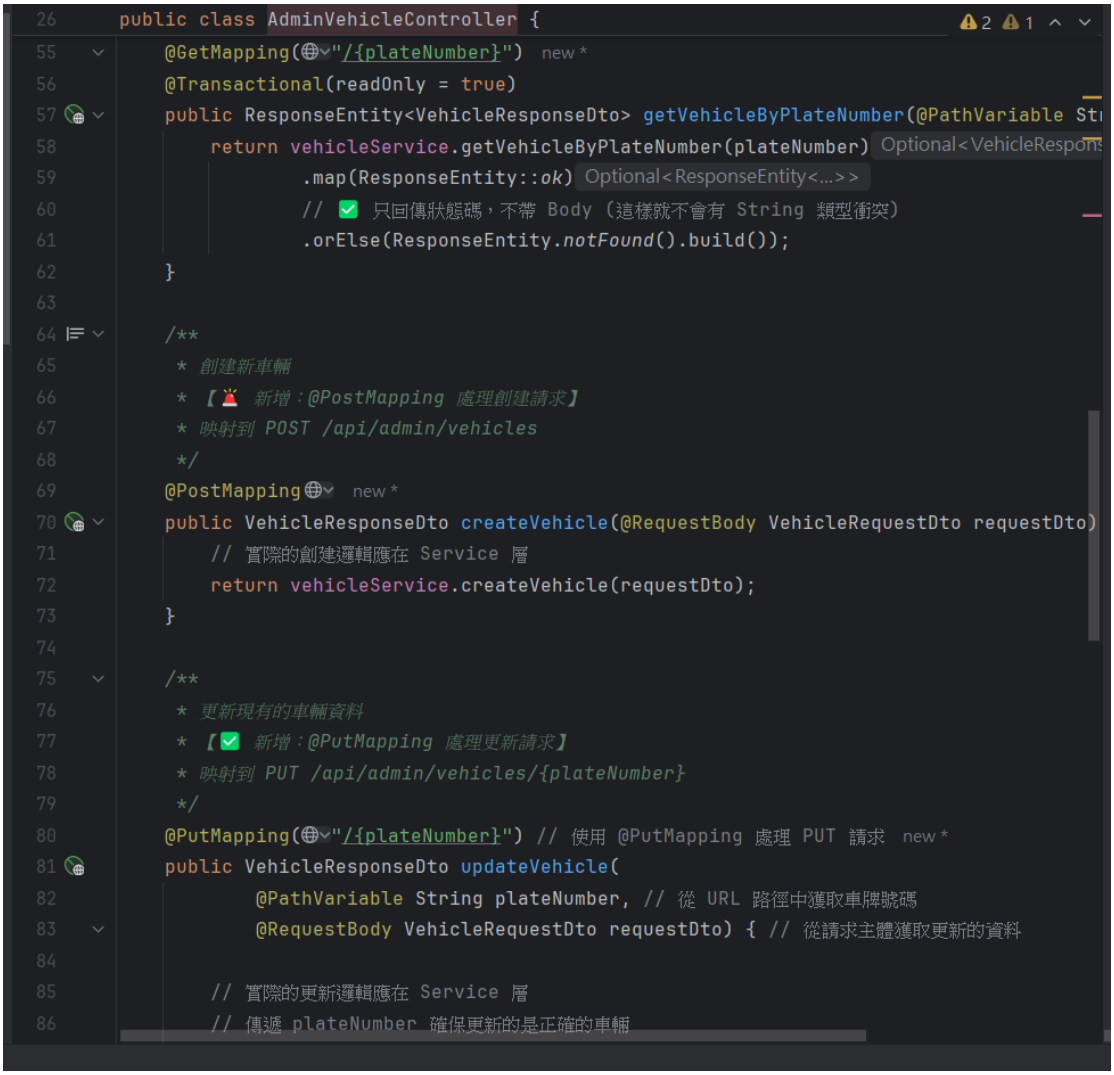
- **級聯操作 (Cascading)**：在 Application 與 ApplicationItem 之間實作 CascadeType.ALL，確保申請單狀態變更時，細項家具資訊能同步維護。

```
21    public class Application {                                                    9
82        @OneToMany(mappedBy = "application", cascade = CascadeType.ALL, orphanRemoval = true)
83        private List<ApplicationItem> applicationItems;
84
```

2. 控制器層級之複雜查詢實作

- **多維度組合查詢**：AdminApplicationController 支援關鍵字、狀態與日期範圍的複合過濾；AdminVehicleController 管理清運車輛資源，並實作 DTO 轉換邏輯，優化前端管理介面的載入效能。





```
public class AdminVehicleController {
    @GetMapping("/{plateNumber}")
    @Transactional(readOnly = true)
    public ResponseEntity<VehicleResponseDto> getVehicleByPlateNumber(@PathVariable St
        return vehicleService.getVehicleByPlateNumber(plateNumber)
                .map(ResponseEntity::ok)
                // ✅ 只回傳狀態碼，不帶 Body（這樣就不會有 String 類型衝突）
                .orElse(ResponseEntity.notFound().build());
    }

    /**
     * 創建新車輛
     * 【🚨 新增：@PostMapping 處理創建請求】
     * 映射到 POST /api/admin/vehicles
     */
    @PostMapping
    public VehicleResponseDto createVehicle(@RequestBody VehicleRequestDto requestDto)
        // 實際的創建邏輯應在 Service 層
        return vehicleService.createVehicle(requestDto);
    }

    /**
     * 更新現有的車輛資料
     * 【✅ 新增：@PutMapping 處理更新請求】
     * 映射到 PUT /api/admin/vehicles/{plateNumber}
     */
    @PutMapping("/{plateNumber}") // 使用 @PutMapping 處理 PUT 請求
    public VehicleResponseDto updateVehicle(
            @PathVariable String plateNumber, // 從 URL 路徑中獲取車牌號碼
            @RequestBody VehicleRequestDto requestDto) { // 從請求主體獲取更新的資料

        // 實際的更新邏輯應在 Service 層
        // 傳遞 plateNumber 確保更新的是正確的車輛
```

- **數據格式規範**：透過 @DateTimeFormat 處理 ISO 日期解析，提升系統對日期字串的健壯性。

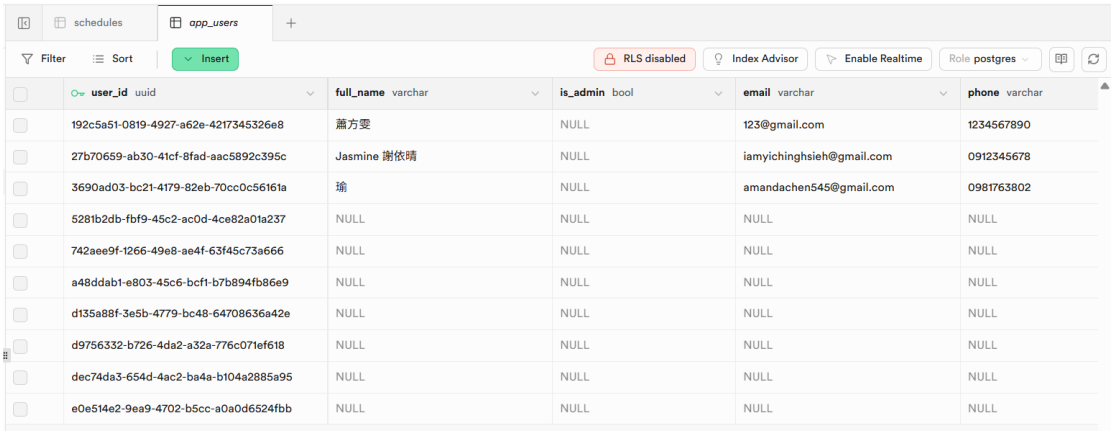
```
public class AdminApplicationController {

    /**
     * GET /api/admin/applications?q=...&status=...&start=...
     * 取得所有狀態的申請列表（支持複雜查詢）
     */
    @GetMapping
    public ApplicationQueryResult getAllApplications(
            @RequestParam(required = false) String q, // 查詢關鍵字（編號/姓名）
            @RequestParam(required = false) String status, // 狀態
            @RequestParam(required = false) @DateTimeFormat(iso = DateTimeFormat.ISO.DATE) LocalDate start, // 開始日期
            @RequestParam(required = false) @DateTimeFormat(iso = DateTimeFormat.ISO.DATE) LocalDate end, // 結束日期
            @RequestParam(defaultValue = "0") int page,
            @RequestParam(defaultValue = "20") int size
    ) {
        return applicationService.searchApplications(q, status, start, end, page, size);
    }
```

第四部分：已知限制、安全性債務與二期計畫

基於開發資源配置與展示定位，本人對系統現存不足進行了技術權衡：

1. 安全性權衡與技術債 (Security Debt)

- **權限存取控制**：目前版本尚未導入自動化權限初始化機制，isAdmin 欄位暫呈 null 狀態。



schedules | app_users

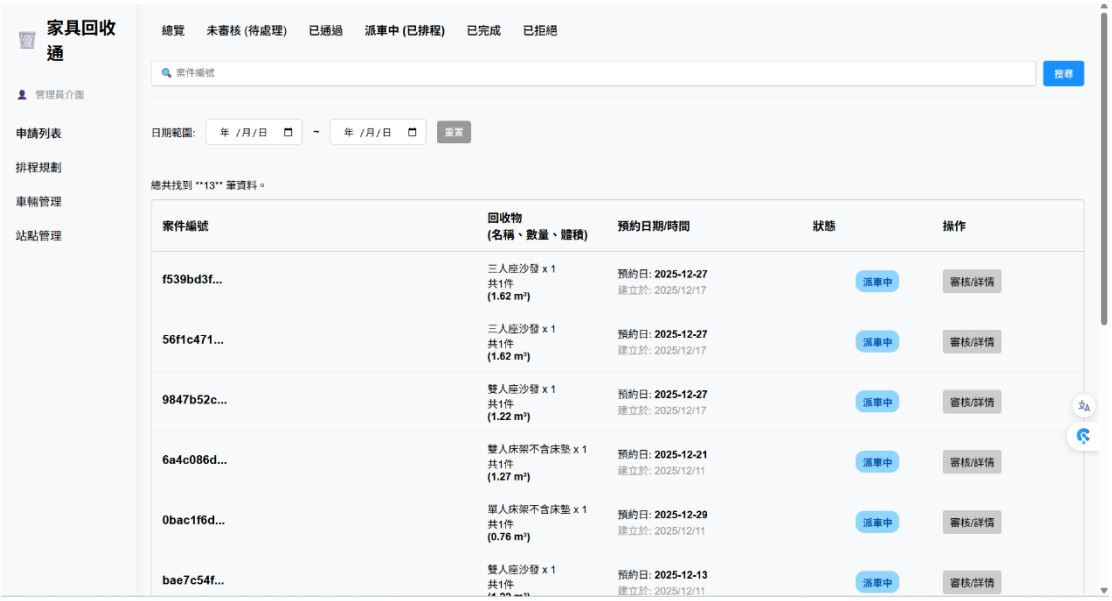
Filter | Sort | Insert | RLS disabled | Index Advisor | Enable Realtime | Role postgres

user_id uuid	full_name varchar	is_admin bool	email varchar	phone varchar
192c5a51-0819-4927-a62e-4217345326e8	蕭方雯	NULL	123@gmail.com	1234567890
27b70659-ab30-41cf-8fad-aac5892c395c	Jasmine 謝依晴	NULL	iamyichinghsieh@gmail.com	0912345678
3690ad03-bc21-4179-82eb-70cc0c56161a	瑜	NULL	amandachen545@gmail.com	0981763802
5281b2db-fbf9-45c2-ac0d-4ce82a01a237	NULL	NULL	NULL	NULL
742aee9f-1266-49e8-ae4f-63f45c73a666	NULL	NULL	NULL	NULL
a48ddab1-e803-45c6-bcf1-b7b894fb86e9	NULL	NULL	NULL	NULL
d135a88f-3e5b-4779-bc48-64708636a42e	NULL	NULL	NULL	NULL
d9756332-b726-4da2-a32a-776c071ef618	NULL	NULL	NULL	NULL
dec74da3-654d-4ac2-ba4a-b104a2885a95	NULL	NULL	NULL	NULL
e0e514e2-9ea9-4702-b5cc-a0a0d6524fbb	NULL	NULL	NULL	NULL

- **已知風險評估**：系統目前依賴前端路由與 URL 遮蔽隔離，確實存在 API 被直接存取的風險。本人已在後續開發之「外語電商平台」中，完整實作 **JWT 認證機制** 進行補強。

2. 業務邏輯缺失與改善方向

- **物流資訊透明度優化**：目前管理端雖能辨識案件為「派車中」，但無法直觀顯示執行該案之司機姓名與車牌。



家具回收通

管理員介面

申請列表

排程規劃

車輛管理

站點管理

總覽 | 未審核 (待處理) | 已通過 | 派車中 (已排程) | 已完成 | 已拒絕

案件編號 | 搜尋

日期範圍: 年 /月/日 ~ 年 /月/日 | 重置

總共找到 **13** 筆資料。

案件編號	回收物 (名稱、數量、體積)	預約日期/時間	狀態	操作
f539bd3f...	三人座沙發 x 1 共1件 (1.62 m³)	預約日: 2025-12-27 建立於: 2025/12/17	派車中	審核/詳情
56f1c471...	三人座沙發 x 1 共1件 (1.62 m³)	預約日: 2025-12-27 建立於: 2025/12/17	派車中	審核/詳情
9847b52c...	雙人座沙發 x 1 共1件 (1.22 m³)	預約日: 2025-12-27 建立於: 2025/12/17	派車中	審核/詳情
6a4c086d...	雙人床架不含床墊 x 1 共1件 (1.27 m³)	預約日: 2025-12-21 建立於: 2025/12/11	派車中	審核/詳情
0bac1f6d...	單人床架不含床墊 x 1 共1件 (0.76 m³)	預約日: 2025-12-29 建立於: 2025/12/11	派車中	審核/詳情
bae7c54f...	雙人座沙發 x 1 共1件	預約日: 2025-12-13 建立於: 2025/12/11	派車中	審核/詳情

- **解決路徑**：預計導入 **Join Fetch** 查詢技術，將 Vehicle 中的司機資訊直接封裝至傳回前端的 DTO 中。

第五部分：後台管理系統功能驗證 (Demonstration)

演示影片技術章節導覽

- 0:00 - 1:40 | **資源管理**：展示 Station 與 Vehicle 實體之讀取與編輯邏輯。
- 1:40 - 3:37 | **核心引擎：排程調度**：演示 @Transactional 保護下的多表連動更新，驗證案件指派流程。
- **影片配樂說明：**採用富有節奏感的幽默音樂，象徵本人在開發過程中遇到問題時能臨危不亂並想辦法理解背後的原因。

YouTube 連結：https://youtu.be/C-W--GVSZ88?si=hEzJyHNdMl1j4MyI