Assessment 1: Car Rental System

DC_UML_EJIlagan.pdf – Assessment 1
Eduardo JR Ilagan

1. ER Diagram:

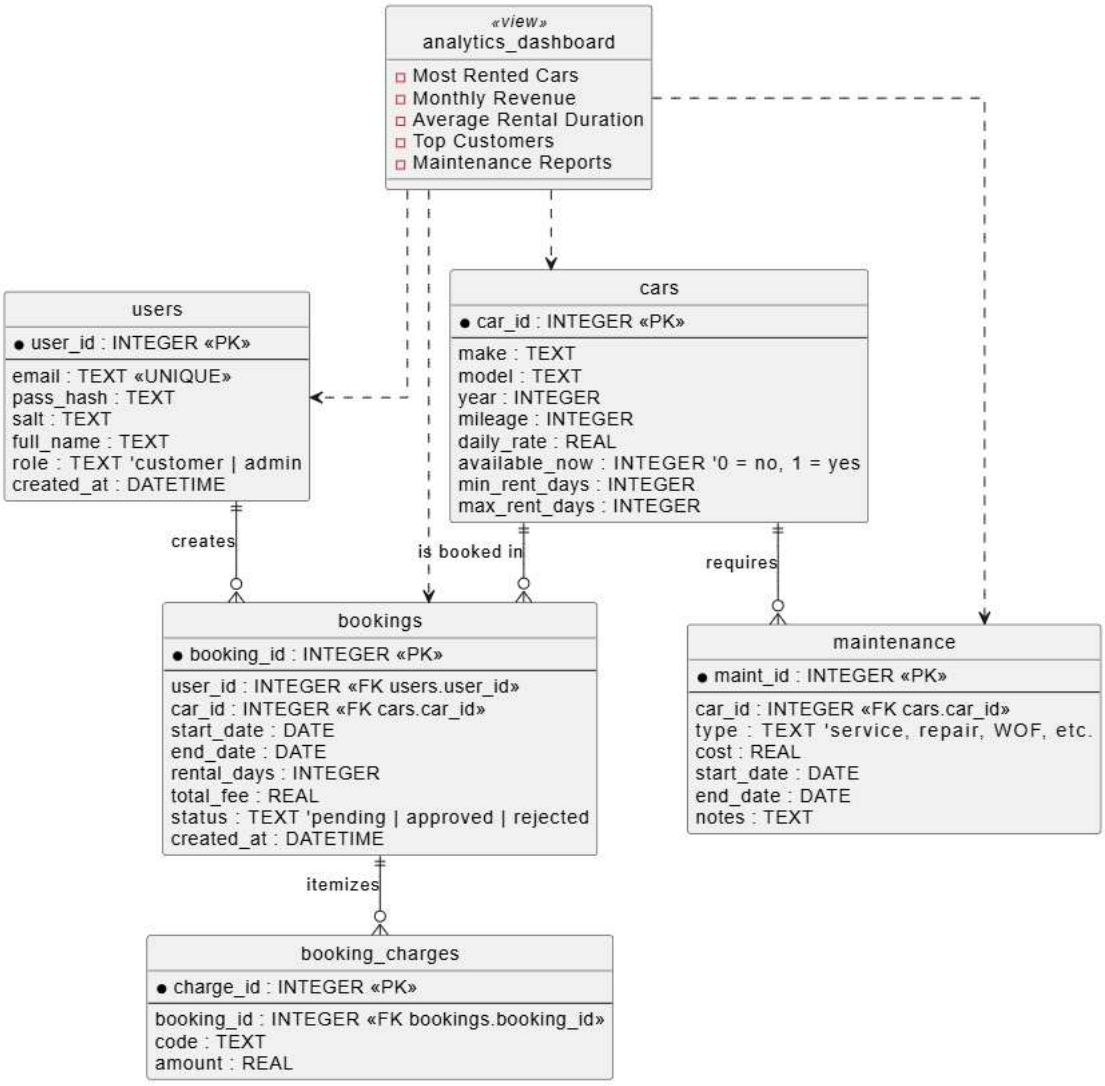


Figure 1. Entity Diagram

Entities

users - Holds identity and access. Each row is a person with a unique email, a hashed+salted password, and a role (customer or admin). Admins can perform

management actions and may create a booking on behalf of a customer; ownership still points to the customer via user_id.

cars - Represents the rentable fleet. Core details (make/model/year/color/mileage) plus rental rules (daily_rate, min_rent_days, max_rent_days) and an operational flag (available_now) for quick checks. True availability is decided by bookings and maintenance windows, not just the flag.

bookings - The reservation record. Links a customer to a car for a date range (start_date..end_date), derives rental_days, and carries lifecycle state (pending, approved, rejected). total_fee is a stored snapshot for invoicing/audit and is explained by its booking_charges. If an admin creates it for a customer, user_id still references the customer.

booking_charges Line-items that make fees transparent. Examples: base rent, weekend uplift, insurance, discounts. Positive and negative amounts are supported; the sum of all charges must equal bookings.total_fee.

maintenance - Service/repair/WOF periods that block a car from being rented. Each record ties to a car, has a start/end date, optional cost and notes. While a maintenance record is open (no end date yet), the car is unavailable; closing the record returns the car to the pool (subject to existing bookings).

analytics_dashboard (view) - A read-only view over the base tables. Surfaces KPIs such as most rented cars, monthly revenue, average duration, top customers, and maintenance reports. No writes; it exists to inform decisions without affecting operational data.

2. Use Cases:

ID	Use Case	Actor	Goal / Description	Functionality
UC-01	Register	Customer	Creates a new customer account with unique email. Password is stored as hash+salt; role defaults to customer. Admin may also register new admins	Authentication
UC-02	Login	Customer / Admin	Authenticates by email/password and issues a session/JWT. Resolves role (customer/admin) for authorization—no data mutation beyond auth state.	Authentication

UC-03	View Available Cars	Customer	Customer searches the fleet by dates and constraints. System filters by car rules (min/max days), existing bookings, and open maintenance windows.	Customer Booking
UC-04	Create Booking	Customer	Customer (or admin on behalf) books a specific car for a valid date range. Calculates fees via include, creates the booking as Pending, and stores line-items if used.	Customer Booking
UC-05	View Booking Status	Customer	Customer checks a booking's state (pending/approved/rejected) and fee snapshot. Read-only; reflects admin decisions and any calculated charges.	Customer Booking
UC-06	Calculate Fees (include)	System (internal)	Internal service. Computes total = base (daily_rate × rental_days) ± extras/discounts; persists itemized charges and reconciles to bookings.total_fee.	Customer Booking
UC-13	Block Car for Maintenance (include)	System (internal)	Effect of an open maintenance window. Car is omitted from availability and cannot be approved for overlapping bookings.	Customer Booking
UC-14	Unblock Car after Maintenance (include)	System (internal)	Effect of completing maintenance. Car re-enters availability; normal booking and approval rules apply.	Customer Booking
UC-15	Check Maintenance Conflicts (include)	System (internal)	Gate inside approval. Prevents Approve if the booking window overlaps any open maintenance for that car.	Customer Booking
UC-07	Manage Cars (Add/Update/Delete)	Admin	Admin maintains the car catalog and rental policy fields (rates, min/max days). Validations enforced; optional uniqueness on make/model/year/color.	Admin Management
UC-08	Approve/Reject Booking	Admin	Admin reviews Pending bookings and decides. Includes a maintenance conflict check; approval reserves the dates, rejection frees the slot.	Admin Management
UC-09	Analytics Dashboard (Reports)	Admin	Read-only KPIs: most-rented cars, monthly revenue, average duration, top customers. Backed by the analytics view; no writes to base tables.	Admin Management

UC-10	Record Maintenance	Admin	Admin opens a maintenance record (service/repair/WOF) for a car. Triggers blocking include so the car is excluded from availability during the window.	Maintenance
UC-11	Complete Maintenance	Admin	Admin closes an active maintenance record by setting end date. Triggers unblocking include so the car returns to the pool (subject to bookings).	Maintenance
UC-12	Maintenance Analytics	Admin	Read-only maintenance KPIs: costs, downtime, frequency, upcoming windows. Powered by the analytics view; operational tables remain unchanged.	Maintenance

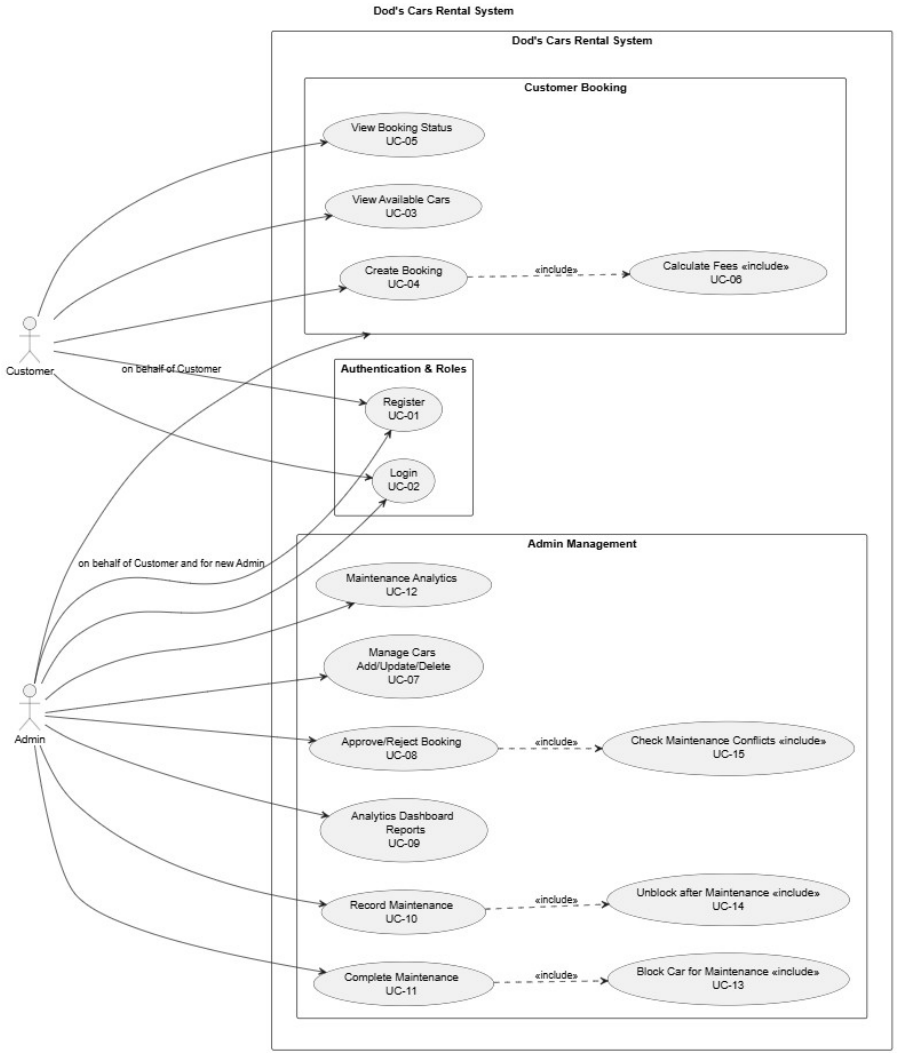
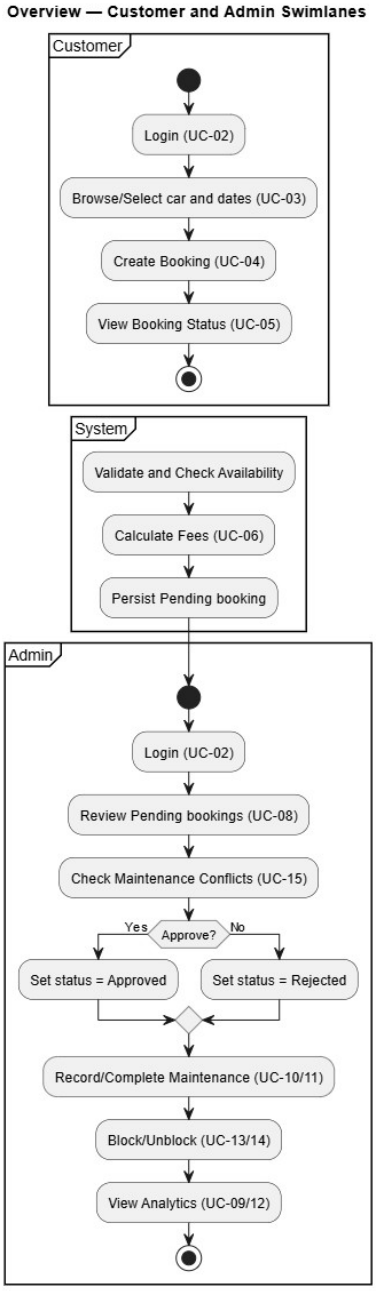


Figure 2. Use Case Diagram

3. Activity Diagrams

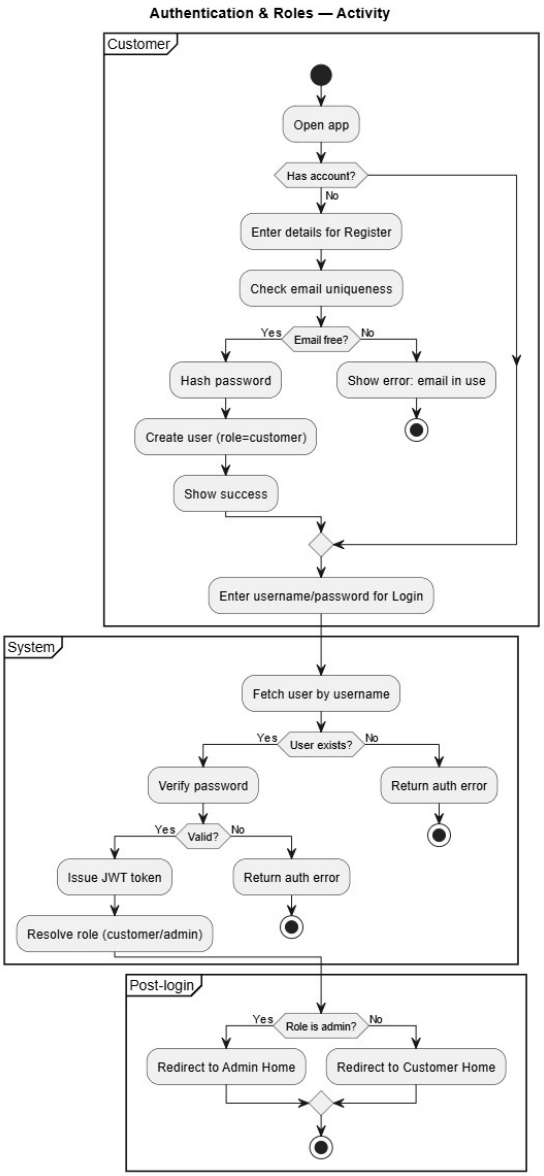
3.1 Overview — Customer & Admin Swimlanes



- Flow Customer: Login → Browse/Select → Create Booking → View Status
- System: Validate → Availability check → Fee calculation → Persist Pending.
- Admin: Login → Review Pending → Conflict check → Approve/Reject.
- Maintenance: Record/Complete impacts availability globally.
- Analytics: Reads from base tables; zero side effects.

Figure 3. Activity Diagram – Overview

3.2 Authentication and Roles



- Customer opens app → chooses Register or Login.
- Register: enter details → system checks email uniqueness → hash+salt → create users row (role=customer) → success.
- Login: enter credentials → system verifies → issues JWT → resolves role.

Figure 4. Activity Diagram – Authentication and Roles

3.3 Admin Management

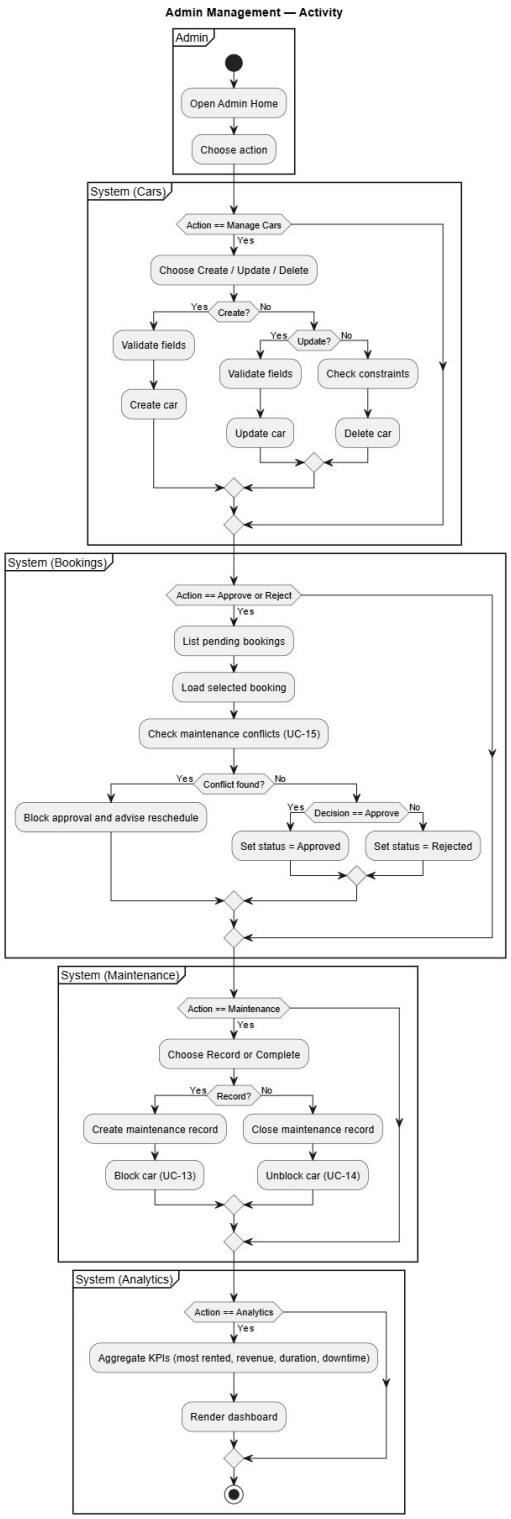


Figure 5. Activity Diagram Admin Management

Branch A — Manage Cars (UC-07)

- Choose Create / Update / Delete.
- Validate fields; apply change to cars.
- Guard deletes with constraints (history protection recommended).

Branch B — Approve/Reject Booking (UC-08 + UC-15 include)

- List Pending bookings → load selected.
- Include: Check Maintenance Conflicts (UC-15) against maintenance.
- Also ensure no approved booking overlap for same car.
- Decision: Approve → set status=approved; Reject → status=rejected.

Branch C — Maintenance lifecycle (UC-10 / UC-11 + UC-13 / UC-14 includes)

- Record Maintenance: INSERT into maintenance (car, start, notes, optional cost). → Effect: Block (include UC-13) during open window.
- Complete Maintenance: set end_date. → Effect: Unblock (include UC-14); car re-enters pool (respect existing bookings).

Branch D — Analytics (UC-09 / UC-12)

- Aggregate KPIs. Read from analytics_dashboard view. No writes.

3.3 Customer Booking

- Browse cars (optional date window).
- System filters: cars rules (min/max), exclude conflicts from bookings + open maintenance.
- Customer selects car + dates.

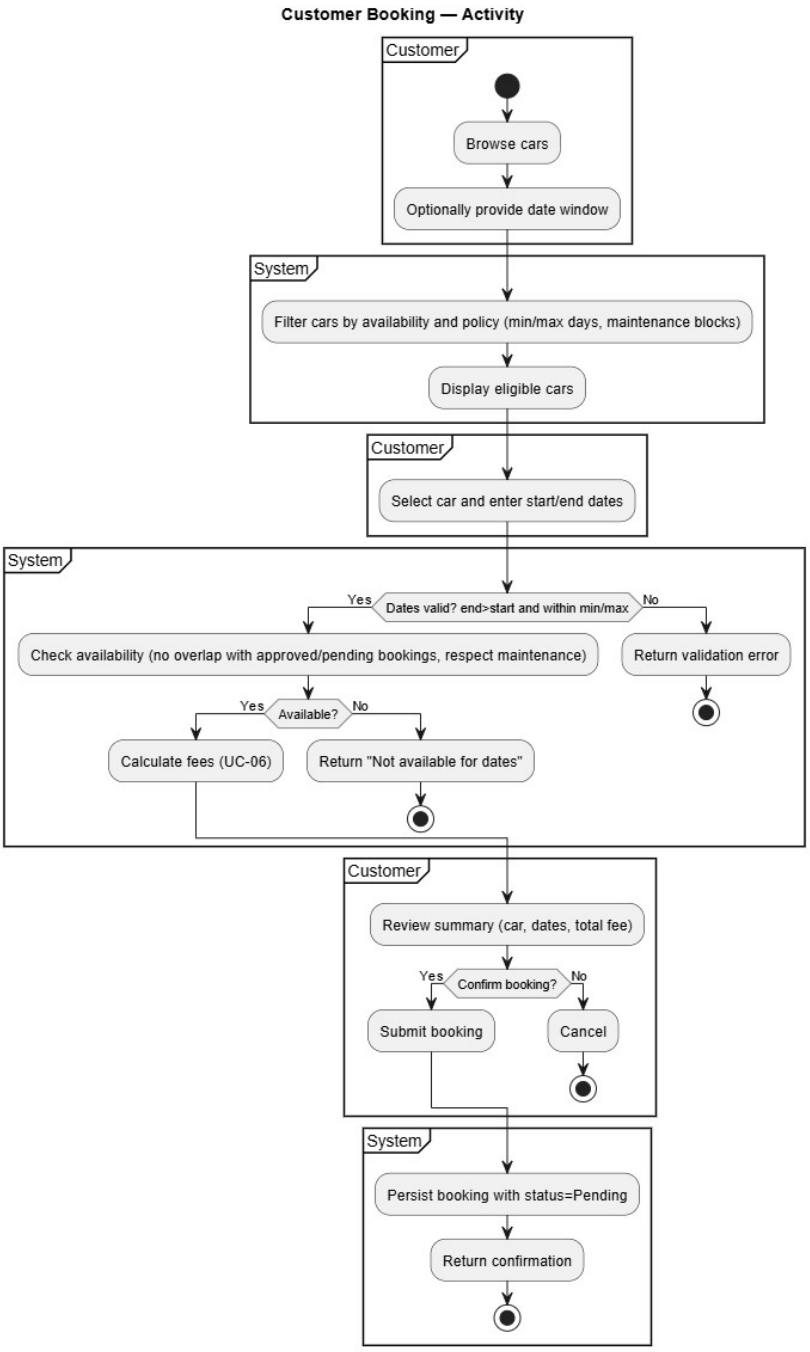


Figure 6. Activity Diagram Customer Booking

- System validates dates and range.
- Include: Calculate Fees (UC-06)
    - Compute rental_days, base = daily_rate × days.
    - Add line items (booking_charges: extras/uplifts/discounts).
    - Reconcile sum → bookings.total_fee.
- Customer reviews summary → confirms.
- System inserts bookings (status=pending, created_at=now).

4. Sequence Diagram

4.1 UC-01 Register

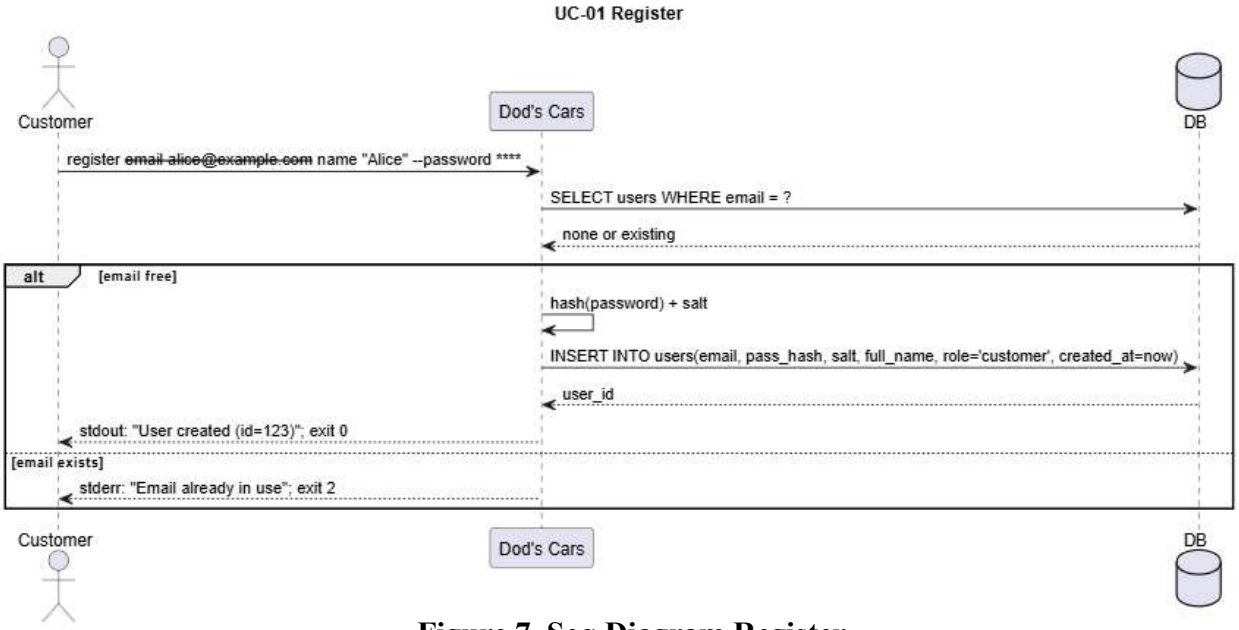


Figure 7. Seq Diagram Register

4.2 UC-02 Login

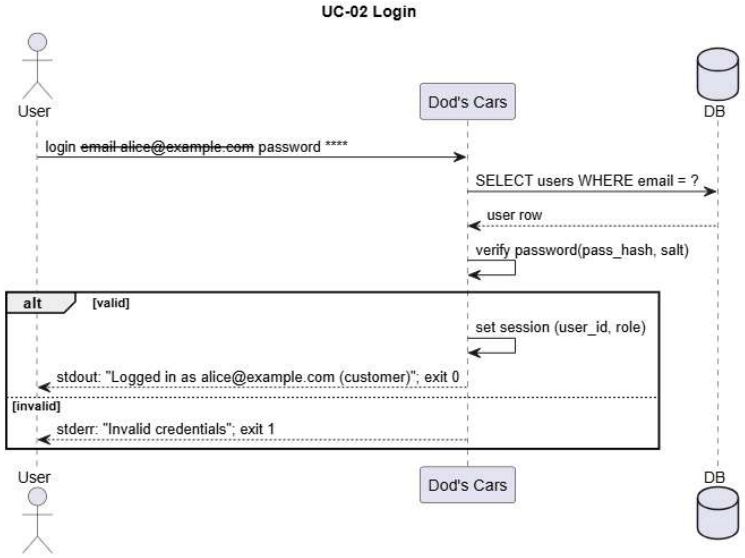


Figure 8. Seq Diagram Login

4.3 UC-03 View A

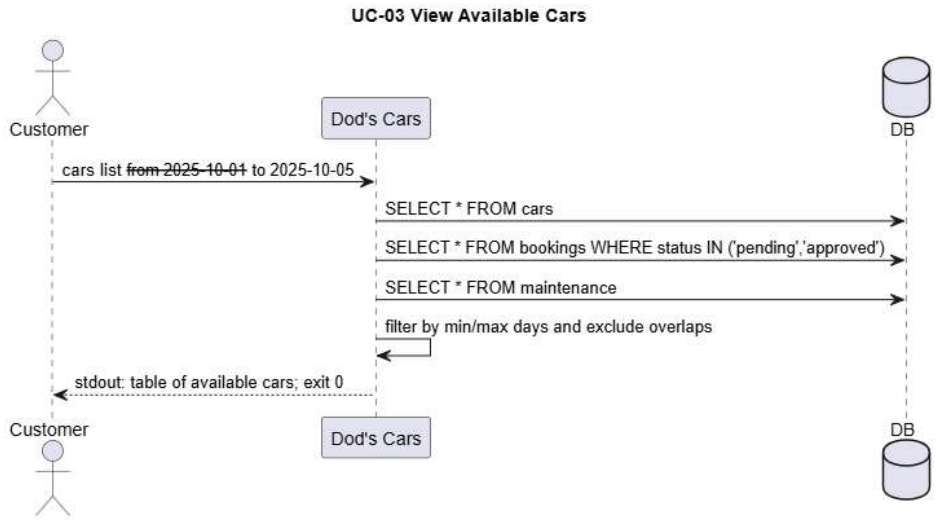


Figure 9. Seq Diagram View Available Cars

4.4 UC-04 Create Booking

Figure 10. Seq Diagram Create Booking

4.5 UC-04 Create Booking on behalf of Customer

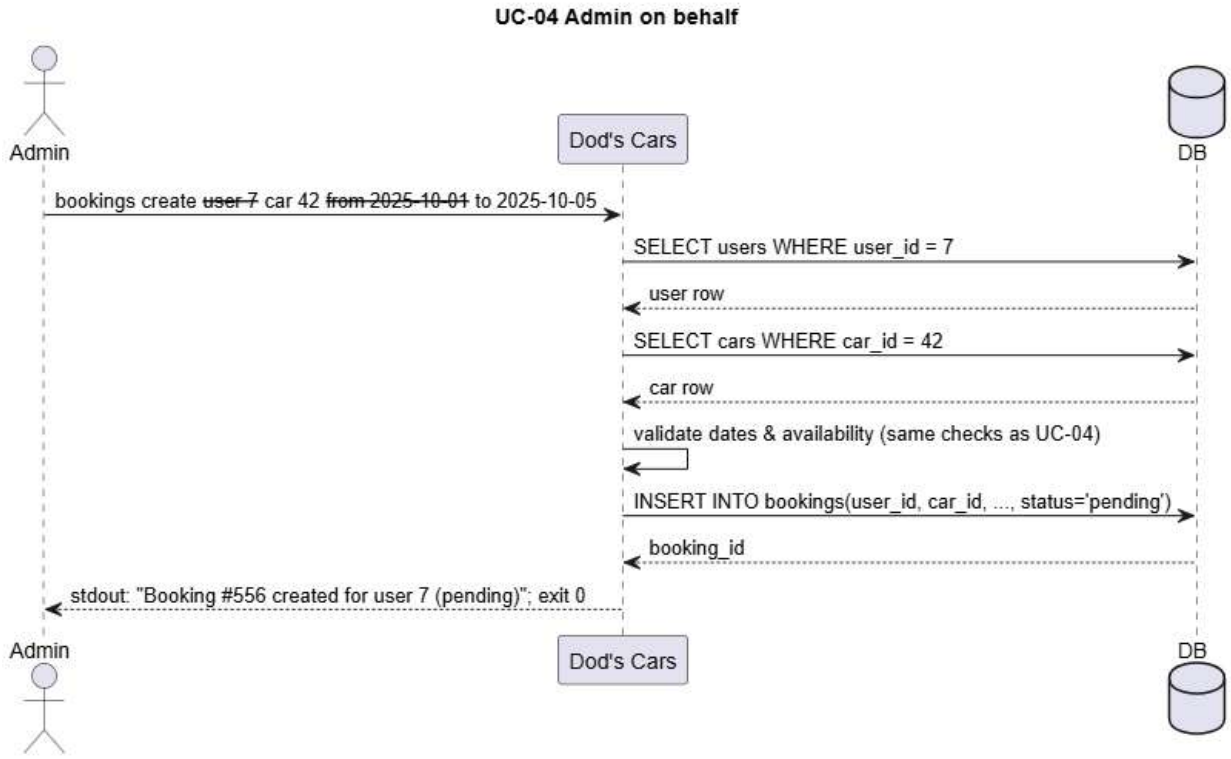


Figure 11. Seq Diagram Create Booking in behalf

4.6 UC-05 View Booking Status

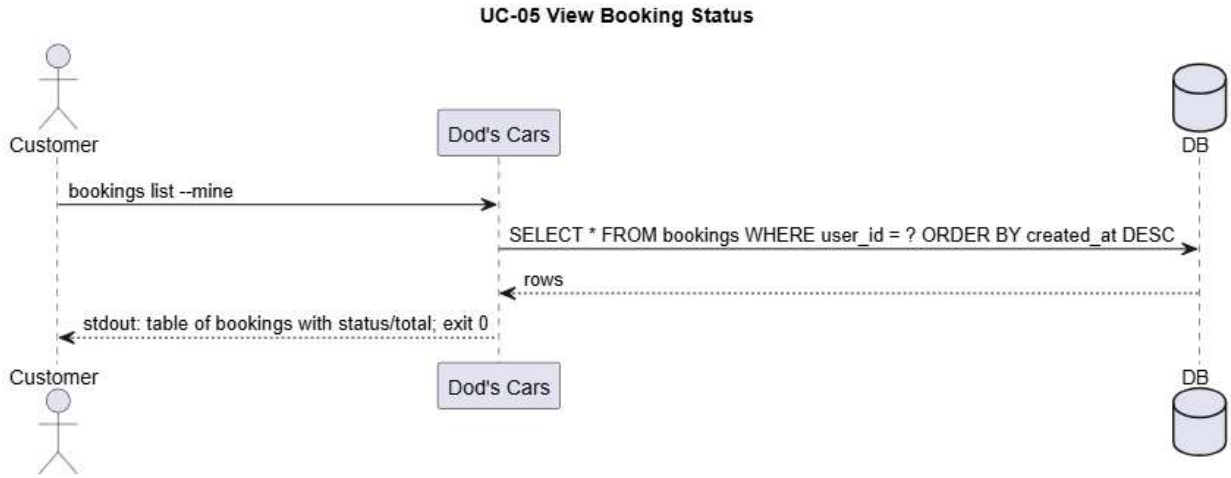


Figure 12. Seq Diagram View Booking Status

4.7 UC-06 Manage Cars (Add/Update/Delete)

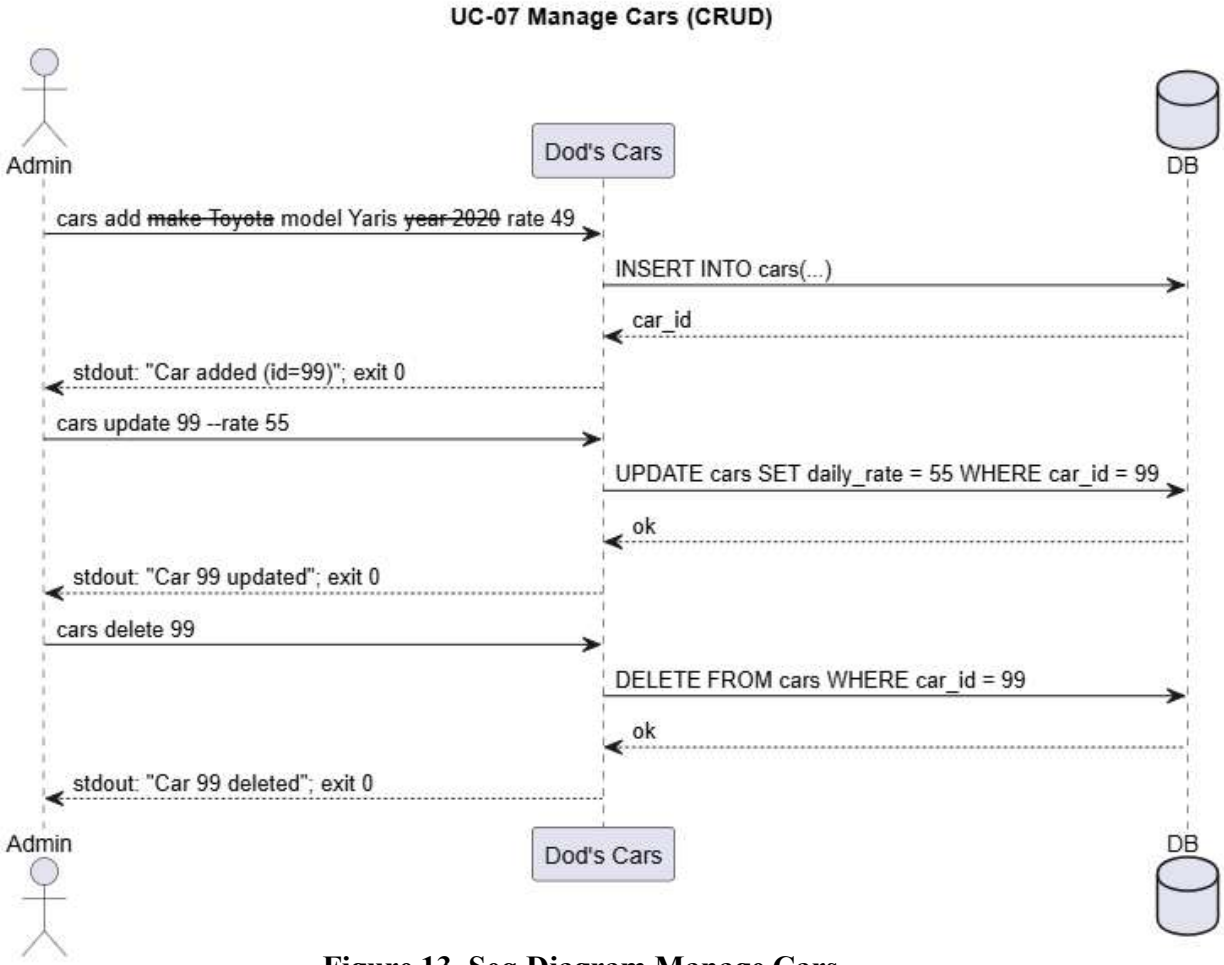


Figure 13. Seq Diagram Manage Cars

4.8 UC-08 + UC-15 Approve/Reject Booking

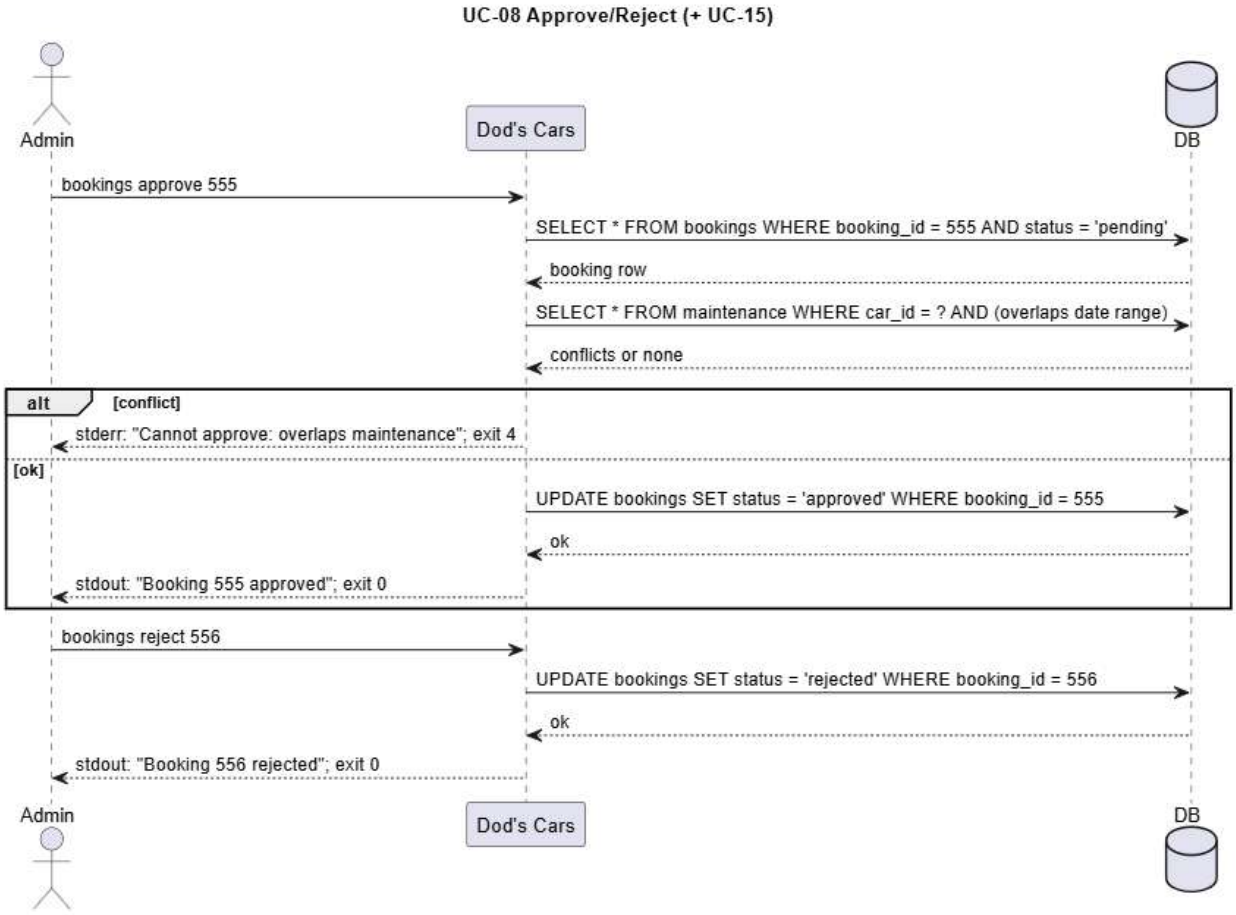


Figure 14. Seq Diagram Approve Reject

4.9 UC-14 Analytics Dashboard (Reports)

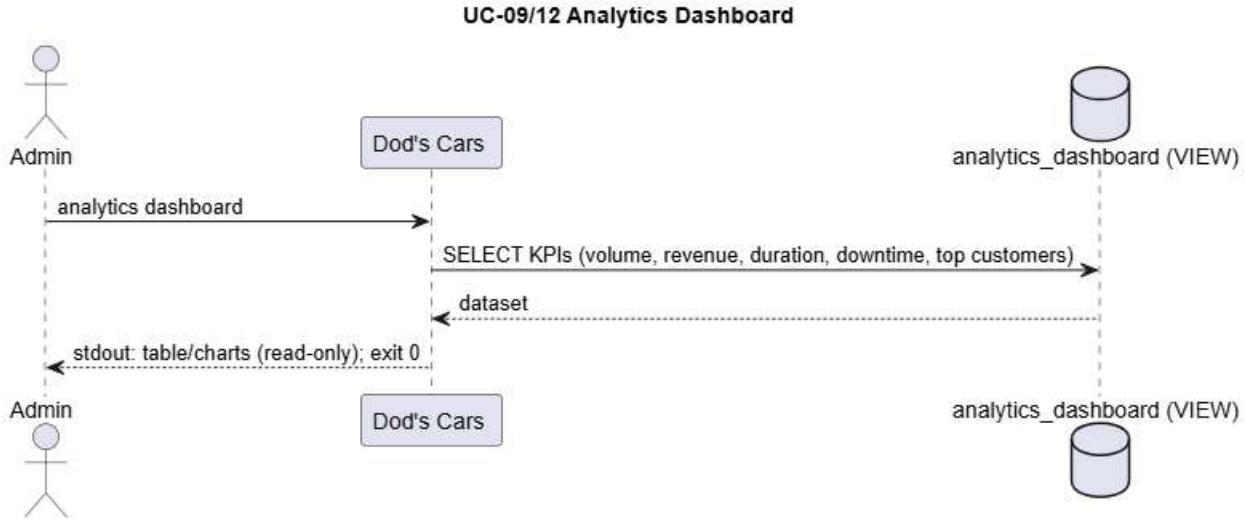


Figure 15. Seq Diagram Analytics Dashboard

4.1 UC-10 Record Maintenance

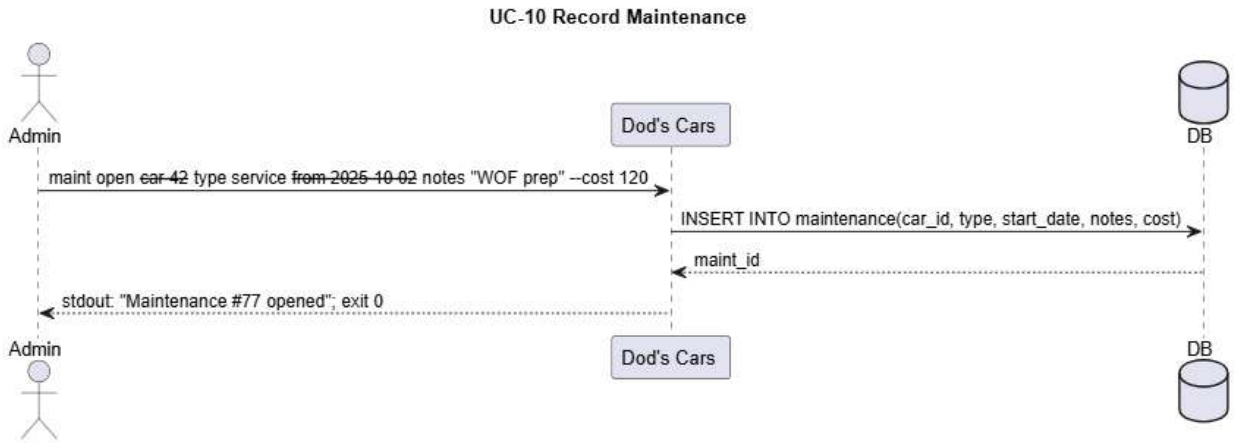


Figure 16. Seq Diagram Record Maintenance

4.11 UC-07 Complete Maintenance

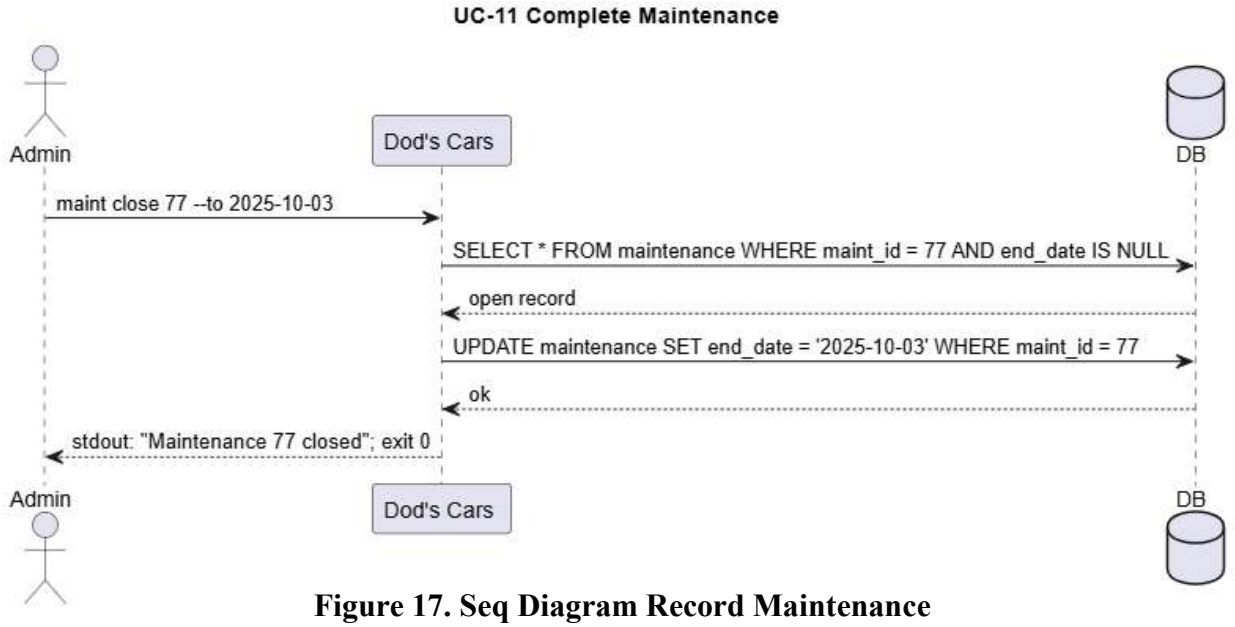


Figure 17. Seq Diagram Record Maintenance

5. Class Diagrams

5.1 Domain Core (Entities)

- **What it shows:** The data model you persist.
- **Key classes:** User, Car, Bk (booking), Chg (charge), Maint (maintenance) + enums Role, BkStatus.
- **Relationships:** User 1–* Bk, Car 1–* Bk, Bk 1–* Chg, Car 1–* Maint.
- **Notes:** Short attrs map to ERD (e.g., Bk.total → bookings.total_fee, Car.rate → cars.daily_rate).

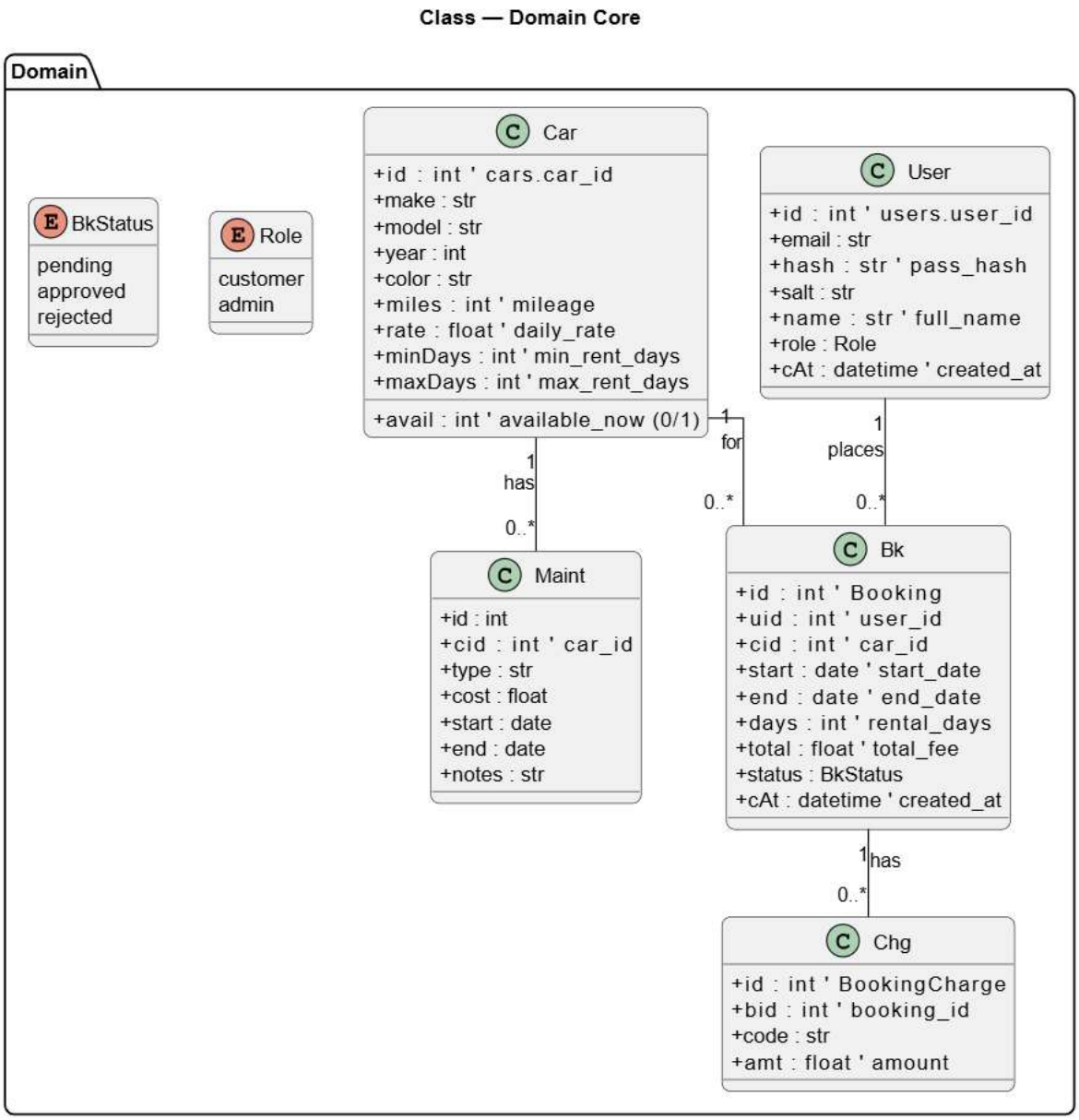


Figure 18. Class Diagram – Domain Core

5.2 **Use-Case Services**

- **What it shows:** Business actions and who they depend on.
- **Services:** AuthSvc (register/login), AvailSvc (availability), PriceSvc (charges/total), BookSvc (create/list/approve/reject), MaintSvc (open/close), AnalytSvc (dashboard).
- **Deps:** Services call repos (URepo, CRepo, BkRepo, ChgRepo, MRepo) and each other (e.g., BookSvc → AvailSvc, PriceSvc).

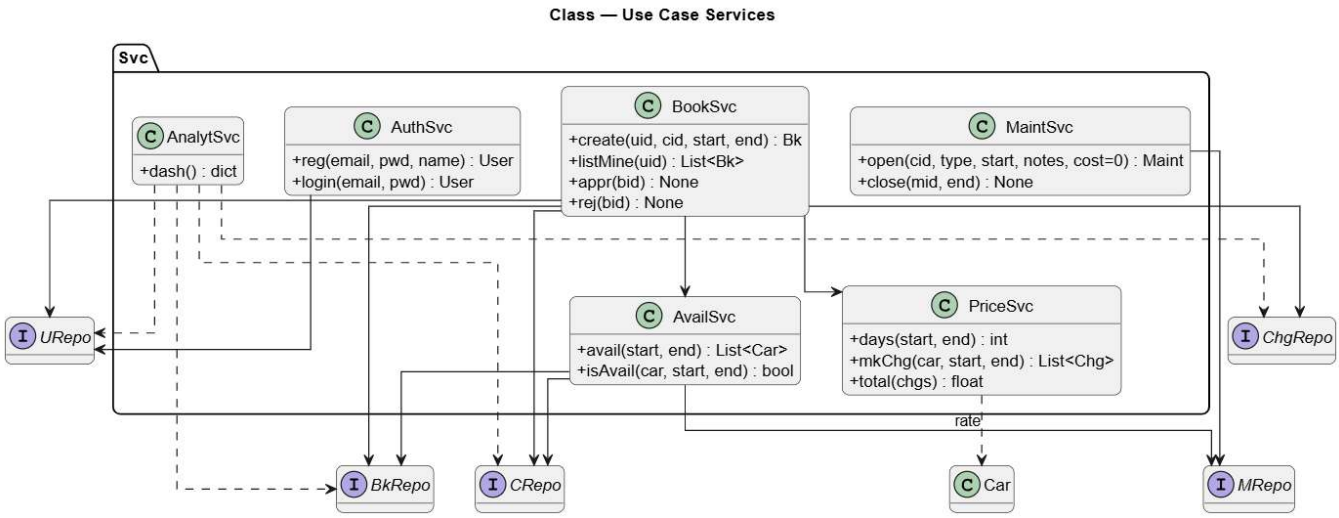


Figure 19. Class Diagram – Use Case Services

5.4 Persistence (SQLite)

- **What it shows:** DB-facing interfaces per table + unit of work.
- **Repos:** URepo, CRepo, BkRepo, ChgRepo, MRepo with compact ops (byId, add, upd, del, setStatus, overlaps).
- **Unit of Work:** SqlUoW provides commit/rollback; repos depend on it for atomic writes (e.g., booking + charges).

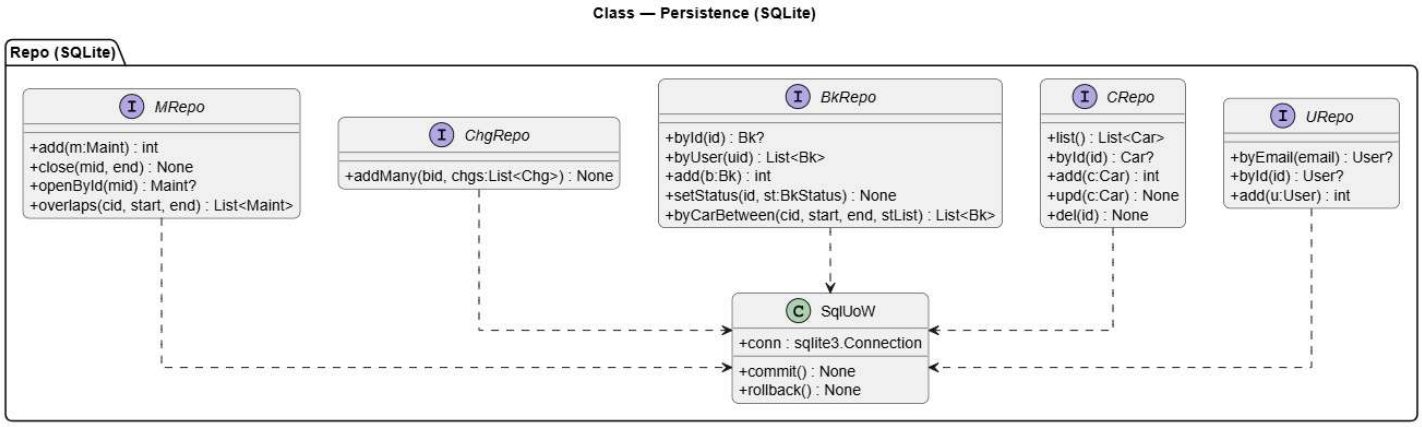


Figure 20. Class Diagram – Class Persistence (SQLite)

5.5 **CLI Orchestration**

- **What it shows:** CLI commands and which service they hit.
- **CLI methods:** reg, login, cars, book, my, appr, rej, mOpen, mClose, dash.
- **Flow:** Thin CLI → service calls; aligns with your CLI sequence diagrams

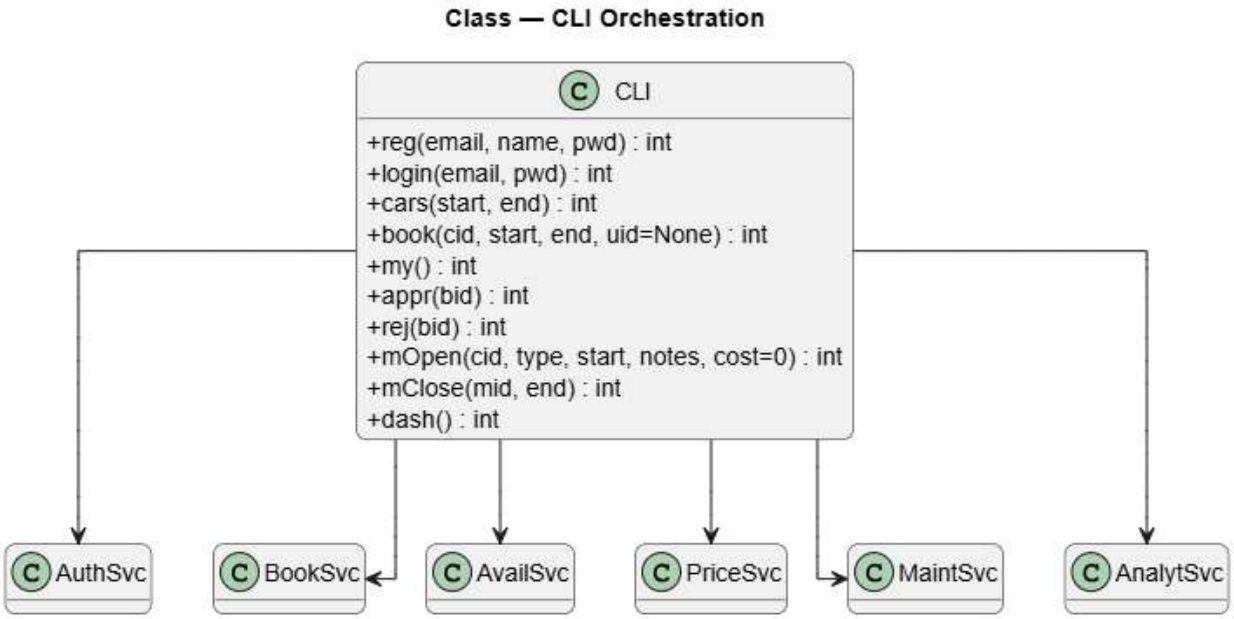


Figure 21. Class Diagram – CLI Orchestration