TAXITUR.NO — Full Application Flow & Post-Implementation Report

Project: TAXITUR / DVBS (CodeIgniter) · Local: http://localhost/txn · Live: www.taxitur.no
Report date: 12 August 2026 (updated) · All roles, marketplace flow, RBAC, GDPR, payments, and completed fixes

> **Current platform status:** Online booking (name plate, special requests, multi-stop, day trip/sightseeing, km+minute+hour pricing) is live. Stripe return, live tracking, owner document pack, admin minute/hour rates, and booking detail fields are implemented. **Role-based access control (RBAC) and GDPR isolation are now enforced:** only Admin has full access; Car Owner, Corporate, Driver, and Customer each see and manage only their own records.

1. Implementation summary

Area	Change	Status
Online booking	Trip type, name plate, special requests, 4 stops + wait hours, km+minute+hour pricing, category UI	**DONE**
Stripe return	Valid `is_conformed=confirm`; booking saved on success	**DONE**
Live tracking	`users/track/{ref}` + driver `update_location`	**DONE**
Car owner docs	Profile & document pack + bank fields at `/carowners/profile`	**DONE**
Admin pricing	`price_per_minute` + `price_per_hour` on vehicle categories	**DONE**
RBAC Admin-only full access	Admin panel blocked for car owners; settings/reports admin-only	**DONE**
RBAC Car owner scope	Own vehicles (`added_by`), own drivers (`created_by` + vehicle assignment), own fleet bookings & payments	**DONE**
RBAC Corporate scope	Corporate portal corporate-only; bookings/branches/invoices scoped by `corporate_id`	**DONE**
GDPR Driver isolation	Calendar, details, cancel, extra charges, customer view — own `driver_id` only; nearby jobs redacted	**DONE**
Owner accepted bookings	`carowners/get_bookings?filter=accepted` for confirm/accept/on_the_way/trip_started/completed	**DONE**
Atomic dispatch + category match	First-wins accept race-safe filter	**PARTIAL**
Public car-owner self-registration	Still admin-created	**PARTIAL**
Admin payout/settlement UI	Bank data collected; payout workflow next	**PARTIAL**

2. Access control model (who sees what)

> **Rule:** Admin = all records. Every other role = own records only (GDPR / data minimisation).

Role	Group	Portal	Data scope	Can manage
Administrator	admin (1)	/admin	All bookings, users, vehicles, settings, reports, finances	Full CRUD on every module
Car Owner	company (5)	/carowners	Vehicles where `added_by` = owner; drivers where `created_by` = owner or assigned to owner vehicles; bookings where `driver_id` ∈ owner fleet	Cars, drivers, accepted/assigned bookings, payments, salaries, documents
Corporate	corporate (7)	/corporate/*	Bookings where `corporate_id` = company; own branches & invoices	Company profile, branches, book trips, view/export own bookings
Driver	driver (4)	/driver	Bookings where `driver_id` = self; nearby unassigned jobs (PII redacted until accept)	Accept job, start/stop/complete own trips, GPS, own profile
Executive	executives (3)	/executive	Operational bookings (verify/confirm workflow)	Dashboard, pending verification — not admin settings
Customer	members (2) / guest	/users/* + public booking	Bookings where `user_id` = self (or guest email match)	Book, pay, track own trip, view own history

Demo login (local): admin@admin.com / admin123 · Login: http://localhost/txn/en/auth/login

3. End-to-end marketplace flow

A. Customer books online

1. Home → pickup & drop-off (airport/local toggles).
2. Trip type: Transfer / Day trip / Sightseeing.
3. Transfer: name plate (airport/hotel/cruise/train).
4. Up to 4 stops + waiting hours; return journey optional.
5. Special needs free text.
6. Maps → distance + journey minutes → fare = start + km + minutes + stop wait + day-trip hours + night.
7. Passenger details → Payment (cash / Stripe / PayPal / PayU).
8. Booking created `pending` or `confirm` (after payment).

B. Dispatch to drivers (GDPR-safe)

1. Drivers with GPS see nearby **unassigned** jobs within `dispatch_radius` (default 100 km).
2. Phone/email hidden on nearby list until driver accepts (GDPR).
3. Accept → `driver_id` set; status: accept → on_the_way → trip_started → completed.
4. Driver `update_location` pushes GPS to active booking for customer live map.
5. Car owners see nearby unassigned jobs (redacted) + all bookings assigned to their drivers.

C. Customer live tracking

1. Booking details → "Open live map" → `users/track/{ref}`.
2. Polls `users/track_location/{ref}` every 10s.
3. Map: pickup, drop-off, moving driver marker (own booking only).

D. Role portals (scoped access)

1. **Admin** — all modules, manual bookings, settings, global reports.
2. **Car owner** — fleet vehicles, own drivers, assigned/accepted bookings, payments, document pack, driver salaries.
3. **Corporate** — register/login, branches, book & view company trips, CSV invoice export (own company only).
4. **Driver** — nearby jobs, own calendar/history, accept & complete own trips only.

4. Role-by-role workflow & access (detailed)

4.1 Customer

Action	Route / module	Access rule
Book & pay online	Home → Welcome/Bookingsystem	Public / logged-in member
View booking history	/users/*	`user_id` = logged-in user
Live track trip	/users/track/{ref}	Own booking ref only
View booking details	/users/booking_details	Own booking only

4.2 Driver GDPR ISOLATED

Action	Route	Access rule
Dashboard stats	/driver	Aggregates for own `driver_id` only
Booking history AJAX	/driver/ajax_get_bookings	`WHERE driver_id = me`
Calendar	/driver/bookingCalendarData	Own trips only (was all trips — fixed)
Booking details	/driver/bookingDetails/{id}	Own `driver_id` only
Nearby jobs	/driver/get_nearby_bookings	Unassigned + redacted phone/email
Accept / update status	/driver/update_booking_status	Own booking or unassigned claim
Complete ride / extra charges	/driver/complete_ride, update_extra_charges	Own `driver_id` required
View customer	/driver/userDetails/{id}	Only if customer linked to own assigned booking
Profile edit	/driver/profile	Own user ID only
GPS push	/driver/update_location	Updates own user + own active bookings

4.3 Car Owner FLEET SCOPED

Action	Route	Access rule
Manage vehicles	/vehicles, /vehicle_settings	`added_by` = owner; edit/delete guarded
Manage drivers	/carowners/drivers_index, /drivers	`created_by` or vehicle assignment
Add driver	/carowners/add_driver	Sets `created_by` on new driver user
View bookings	/bookings (car template)	Bookings for owner fleet drivers only
Accepted bookings API	/carowners/get_bookings?filter=accepted	confirm/accept/on_the_way/trip_started/completed
Payments	/carowners/payments	Owner fleet bookings only
Nearby dispatch view	/carowners/get_nearby_bookings	Unassigned (redacted) + own fleet assigned
Documents & bank	/carowners/profile	Own profile only
Driver salaries	/carowners/driver_salaries	`owner_id` scoped
Admin panel	/admin	**Blocked** — redirected to /carowners

4.4 Corporate COMPANY SCOPED

Action	Route	Access rule
Register / login	/corporate/register, /auth/login	Corporate group
Dashboard	/corporate/dashboard	Own corporate profile
Branches CRUD	/corporate/branches	`corporate_id` match on branch
Book trip	/corporate/new_booking	Saves `corporate_id` on booking
View bookings	/corporate/bookings	`corporate_id` or fallback `user_id`
Booking details	/corporate/booking_details/{id}	Own company booking only
Invoice CSV export	/corporate/invoices_export	Date range + own company only
Admin panel	/admin	Corporate users use /corporate/* only

4.5 Admin & Executive

Role	Scope	Notes
Admin	Full platform	All CRUD, settings, reports, car owner list, global payments
Executive	Operations	Dashboard, verify bookings; reports module; not full settings

5. RBAC helper functions (central enforcement)

Implemented in `application/helpers/global_helper.php` — used across controllers:

Function	Purpose
`require_admin_access()`	Admin-only routes; redirects car owners to /carowners
`check_access('car_owner')`	Car owner routes (alias for company group)
`get_owner_driver_ids()`	Drivers owned via `created_by` + vehicle assignment
`get_owner_vehicle_ids()`	Vehicles where `added_by` = owner
`append_role_booking_scope()`	Filters booking lists by role (owner/corporate/customer)
`user_can_access_booking()`	Single booking access check for view/edit/delete
`user_can_access_driver()`	Owner can only edit/delete own drivers
`user_can_access_vehicle()`	Owner can only edit/delete own vehicles
`get_corporate_account_id()`	Resolve corporate profile from logged-in user
`redact_booking_customer_pii()`	Hide phone/email on unassigned jobs (GDPR)
`driver_can_view_booking()`	Driver: own assigned or redacted unassigned

6. GDPR & security controls

Control	Implementation	Status
Driver cannot see other drivers' trips	All driver queries filter `driver_id`	**ENFORCED**
Driver calendar was leaking all bookings	Calendar SQL scoped to own driver	**FIXED**
Driver could open any booking by ref/ID	bookingDetails, booking_details require own driver	**FIXED**
Driver could view any customer profile	userDetails only if customer on own booking	**FIXED**
Nearby jobs expose PII before accept	phone/email redacted until assigned	**ENFORCED**
Owner payments showed all platform bookings	Filtered by owner fleet `driver_id`	**FIXED**
Owner saw all drivers in system	Scoped by `created_by` + assignment	**FIXED**
Owner accessed admin financial dashboard	Admin constructor blocks car owners	**FIXED**
Corporate cross-company booking access	`corporate_id` on bookings + query filter	**ENFORCED**
Customer tracking other users' trips	Track/history scoped to own user/booking	**ENFORCED**

7. Pricing formula

> **Fare** = start_fee + (km × price_per_km OR day_rate) + (journey_minutes × price_per_minute) + (stop_wait × price_per_minute) + (duration_hours × price_per_hour) + night segments

8. Database schema (marketplace + RBAC)

Table	Columns / purpose
vbs_bookings	trip_type, pickup_name_plate, special_requests, duration_hours, journey_minutes, waypoints_json, stop_waiting_cost, driver_lat/lng, **corporate_id**, **corporate_branch_id**
vbs_vehicle_categories	price_per_minute, price_per_hour
vbs_vehicle	added_by (car owner link)
vbs_users	created_by (driver → owner link), owner doc/bank fields
vbs_assign_cars_driver	vehicle_id + driver_id (fleet assignment)
vbs_site_settings	dispatch_radius (default 100 km)

Migrations: `sql/booking_enhancements.sql`, `sql/rbac_owner_scope.sql`, `sql/rbac_corporate_bookings.sql`

9. Key URLs (local)

URL	Role	Purpose
http://localhost/txn/	Public	Online booking home
http://localhost/txn/en/auth/login	All	Login
http://localhost/txn/admin	Admin	Full admin dashboard
http://localhost/txn/carowners	Car owner	Fleet portal (scoped)
http://localhost/txn/carowners/profile	Car owner	Documents & bank
http://localhost/txn/driver	Driver	Driver portal (own data)
http://localhost/txn/corporate/dashboard	Corporate	Company portal
http://localhost/txn/users/track/{ref}	Customer	Live trip map
http://localhost/txn/vehicle_categories/addedit	Admin	Minute/hour rates

10. Module CRUD matrix (admin full access)

Module	C	R	U	D	Non-admin scope
Bookings	Y	Y	Y	Y	Owner/Corp/Driver/Customer: own only
Vehicles	Y	Y	Y	Y	Owner: `added_by`
Drivers (users)	Y	Y	Y	Y	Owner: `created_by`
Vehicle categories	Y	Y	Y	Y	Admin (+ owner read per existing rules)
Settings / Reports	Y	Y	Y	Y	Admin only (exec: reports)
Corporate branches	—	—	—	—	Corporate: own company

11. Acceptance checklist (incl. RBAC / GDPR)

#	Test	Pass?
1	Customer: book transfer with name plate + multi-stop + pay	☐
2	Customer: live map shows driver for own booking only	☐
3	Driver: calendar/history shows only own trips	☐
4	Driver: cannot open another driver's booking by changing URL ID	☐
5	Driver: nearby list hides phone/email until accept	☐
6	Car owner: sees only own vehicles and drivers	☐
7	Car owner: payments/bookings limited to own fleet	☐
8	Car owner: /admin redirects away (no global data)	☐
9	Corporate: sees only own company bookings & branches	☐
10	Admin: full access to all modules and reports	☐

12. Files touched (latest cycle)

- `global_helper.php` — RBAC helpers, GDPR redaction, corporate/owner scopes
- `Admin.php` — admin-only constructor
- `CarOwners.php` — fleet scoping, payments, nearby redaction, driver `created_by`
- `Bookings.php` — role-scoped lists, view/edit/delete guards
- `Drivers.php`, `Vehicles.php` — owner-scoped lists & mutations
- `Vehicle_settings.php` — `added_by` on create, edit guard
- `Driver.php` — GDPR isolation (calendar, details, userDetails, charges)
- `Corporate.php` — corporate-only portal, `corporate_id` booking scope
- `Reports.php` — admin/exec only; export scoped by role
- `sql/rbac_owner_scope.sql`, `sql/rbac_corporate_bookings.sql`

13. Remaining enhancements (optional)

Item	Priority
Atomic first-wins accept + vehicle category match on dispatch	P1
Backfill `created_by` / `added_by` for legacy drivers & vehicles	P1
Public car-owner self-registration	P2
Admin payout/settlement workflow	P2
Corporate contact-person field + priced CSV columns	P2