

# Transfers.ma

## Transportation Booking Platform

### Technical Documentation

Django 5.0 | PostgreSQL | Redis | Celery

REST API | JWT Authentication | Stripe/PayPal

# Table of Contents

---

1. Project Overview
2. Technology Stack
3. Architecture & Structure
4. Django Applications
5. Data Models
6. Database Schema
7. API Endpoints
8. Authentication
9. Payment Integration
10. Dashboard Features
11. Deployment

# 1. Project Overview

Transfers.ma is a comprehensive transportation booking and fleet management platform designed for a Morocco-based transportation service company. The platform handles multiple transportation services including airport transfers, point-to-point transfers, tour packages, and car rentals.

## Business Problems Solved

- **Airport & Point-to-Point Transfers:** Ground transportation bookings
- **Multi-Service Offering:** Transfers, tours, and car rentals
- **Fleet Management:** Vehicles, drivers, pricing, and routes
- **Payment Processing:** Stripe and PayPal integration
- **Dynamic Pricing:** Route-based and zone-based pricing
- **Admin Dashboard:** Comprehensive management interface

## Key Features

- Multi-language support (EN, FR, AR, ES)
- Real-time distance calculation via Google Maps API
- JWT-based authentication
- Async task processing with Celery
- RESTful API with OpenAPI documentation

# 2. Technology Stack

Component	Technology
Framework	Django 5.0.1
API	Django REST Framework 3.14.0
Database	PostgreSQL 16
Cache/Queue	Redis 7
Task Queue	Celery 5.3.4
Authentication	djangorestframework-simplejwt

Payments	Stripe 7.10.0, PayPal REST SDK
API Docs	drf-spectacular (OpenAPI/Swagger)
i18n	django-modeltranslation
Web Server	Gunicorn + Nginx
Containerization	Docker & Docker Compose

## 3. Architecture & Structure

---

### Project Layout

```

transfers.ma/
├── config/                # Django configuration
│   ├── settings/         # Environment-specific settings
│   ├── urls.py           # Main URL router
│   └── celery.py         # Celery configuration
├── apps/                 # Django applications
│   ├── accounts/         # User authentication
│   ├── locations/        # Zones, routes, distance
│   ├── vehicles/         # Fleet management
│   ├── transfers/        # Transfer bookings
│   ├── trips/            # Tour packages
│   ├── rentals/          # Car rentals
│   ├── payments/         # Payment processing
│   └── dashboard/        # Admin interface
├── templates/            # HTML templates
├── static/               # CSS, JS, images
└── media/               # User uploads

```

## 4. Django Applications

---

### accounts

Custom User model with email-based authentication. User roles include Admin, Manager, Driver, and Client. Driver profiles track license information and ratings.

## **locations**

Geographic zones with circular or polygon boundaries. Pre-defined routes between locations. Distance calculation using Google Maps API with Haversine fallback.

## **vehicles**

Vehicle catalog with categories (Sedan, SUV, Van, Minibus, Luxury). Features like WiFi, A/C, leather seats. Image gallery support. Zone-based and route-based pricing.

## **transfers**

Point-to-point transfer bookings with flight tracking. Round-trip support. Extras and add-ons (child seats, meet & greet). Driver assignment workflow with status tracking.

## **trips**

Tour packages with itineraries and schedules. Group and private booking options. Participant tracking with adult/child pricing.

## **rentals**

Car rental bookings with insurance options (Basic, Standard, Premium). Mileage and fuel tracking. Different return location fees. Security deposit management.

## **payments**

Payment gateway abstraction layer. Stripe and PayPal integration with webhooks. Refund processing. Automatic invoice generation.

## **dashboard**

Admin management interface with Bootstrap 5. Statistics and charts. Booking, fleet, and user management. Reports and analytics.

# **5. Data Models**

---

## **User Model (accounts)**

Custom user model using email as primary identifier.

- **Fields:** email, first\_name, last\_name, phone, role, avatar, is\_verified, language
- **Related:** Profile (1:1), DriverProfile (1:1)

## Vehicle Model (vehicles)

Individual vehicles in the fleet.

- **Fields:** name, category, license\_plate, passengers, luggage, status, daily\_rate
- **Related:** VehicleCategory, VehicleFeature (M2M), VehicleImage, VehicleZonePricing

## Transfer Model (transfers)

Core booking model for point-to-point transfers.

- **Fields:** booking\_reference, customer info, pickup/dropoff locations, distance, flight info, pricing, status, vehicle
- **Related:** TransferExtra (M2M via TransferExtraBooking)

## Payment Model (payments)

Generic payment supporting multiple booking types.

- **Fields:** reference, content\_type, object\_id, amount, currency, gateway, status
- **Related:** PaymentGateway, Refund, Invoice

# 6. Database Schema

The database consists of 41 tables organized by Django application. Below are the main tables with their columns and relationships.

## Database Tables Overview

Application	Tables
accounts	accounts_user, accounts_profile, accounts_driverprofile, accounts_sitesettings
vehicles	vehicles_vehicle, vehicles_vehiclecategory, vehicles_vehiclefeature, vehicles_vehicleimage, vehicles_vehiclezonepricing
locations	locations_zone, locations_zonedistancerange, locations_zonepricing, locations_route, locations_vehicleroutepricing

transfers	transfers_transfer, transfers_transferextra, transfers_transferextrabooking
trips	trips_trip, trips_tripbooking, trips_tripschedule, trips_tripimage
rentals	rentals_rental, rentals_rentalextra, rentals_rentalextrabooking, rentals_insuranceoption
payments	payments_payment, payments_paymentgateway, payments_refund, payments_invoice

## accounts\_user

Custom user model with email-based authentication and role management.

Column	Type	Nullable	Description
id	BIGINT	NOT NULL	Primary Key
email	VARCHAR(254)	NOT NULL	Unique email address
password	VARCHAR(128)	NOT NULL	Hashed password
first_name	VARCHAR(150)	NOT NULL	First name
last_name	VARCHAR(150)	NOT NULL	Last name
phone	VARCHAR(20)	NOT NULL	Phone number
role	VARCHAR(20)	NOT NULL	ADMIN, MANAGER, DRIVER, CLIENT
avatar	VARCHAR(100)	NULL	Profile image path
is_verified	BOOLEAN	NOT NULL	Email verification status
language	VARCHAR(10)	NOT NULL	Preferred language
is_superuser	BOOLEAN	NOT NULL	Superuser flag
is_staff	BOOLEAN	NOT NULL	Staff access flag
is_active	BOOLEAN	NOT NULL	Active account flag
date_joined	TIMESTAMPTZ	NOT NULL	Registration date
last_login	TIMESTAMPTZ	NULL	Last login timestamp
created_at	TIMESTAMPTZ	NOT NULL	Record creation
updated_at	TIMESTAMPTZ	NOT NULL	Last update

**Indexes:** email (UNIQUE), id (PK)

**Referenced by:** transfers\_transfer, trips\_tripbooking, rentals\_rental, payments\_payment, accounts\_profile, accounts\_driverprofile

## vehicles\_vehicle

Fleet vehicles with category assignment and pricing.

Column	Type	Nullable	Description
id	BIGINT	NOT NULL	Primary Key

name	VARCHAR(100)	NOT NULL	Vehicle name
<i>category_id</i>	BIGINT	NOT NULL	FK → vehicles_vehiclecategory
license_plate	VARCHAR(20)	NULL	License plate number
year	SMALLINT	NULL	Manufacturing year
color	VARCHAR(50)	NOT NULL	Vehicle color
passengers	SMALLINT	NOT NULL	Max passengers (≥0)
luggage	SMALLINT	NOT NULL	Max luggage (≥0)
status	VARCHAR(20)	NOT NULL	AVAILABLE, IN_USE, MAINTENANCE
service_type	VARCHAR(20)	NOT NULL	TRANSFER or RENTAL
daily_rate	DECIMAL(10,2)	NULL	Daily rental rate
weekly_rate	DECIMAL(10,2)	NULL	Weekly rental rate
supplier_name	VARCHAR(200)	NOT NULL	Supplier name
notes	TEXT	NOT NULL	Internal notes
is_active	BOOLEAN	NOT NULL	Active status
created_at	TIMESTAMPTZ	NOT NULL	Record creation
updated_at	TIMESTAMPTZ	NOT NULL	Last update

**Foreign Keys:** category\_id → vehicles\_vehiclecategory(id)

**Referenced by:** transfers\_transfer, rentals\_rental, vehicles\_vehicleimage, vehicles\_vehiclezonepricing, accounts\_driverprofile

## transfers\_transfer

Main booking model for point-to-point transfers.

Column	Type	Nullable	Description
<b>id</b>	BIGINT	NOT NULL	Primary Key
booking_ref	VARCHAR(20)	NOT NULL	Unique booking reference (TRF-XXXXXXX)
customer_name	VARCHAR(200)	NOT NULL	Customer full name
customer_email	VARCHAR(254)	NOT NULL	Customer email
customer_phone	VARCHAR(20)	NOT NULL	Customer phone
<i>customer_id</i>	BIGINT	NULL	FK → accounts_user (optional)
transfer_type	VARCHAR(20)	NOT NULL	AIRPORT_PICKUP, AIRPORT_DROPOFF, CITY_TO_CITY



pickup_address	TEXT	NOT NULL	Pickup location address
pickup_latitude	DECIMAL(10,7)	NULL	Pickup GPS latitude
pickup_longitude	DECIMAL(10,7)	NULL	Pickup GPS longitude
dropoff_address	TEXT	NOT NULL	Dropoff location address
dropoff_latitude	DECIMAL(10,7)	NULL	Dropoff GPS latitude
dropoff_longitude	DECIMAL(10,7)	NULL	Dropoff GPS longitude
pickup_datetime	TIMESTAMP TZ	NOT NULL	Scheduled pickup time
distance_km	DECIMAL(8,2)	NULL	Calculated distance
duration_minutes	INTEGER	NULL	Estimated duration (≥0)
flight_number	VARCHAR(20)	NOT NULL	Flight number (if applicable)
flight_arrival_time	TIME	NULL	Expected flight arrival
passengers	SMALLINT	NOT NULL	Number of passengers (≥0)
luggage	SMALLINT	NOT NULL	Number of luggage (≥0)
child_seats	SMALLINT	NOT NULL	Child seats needed (≥0)
<i>vehicle_category_id</i>	BIGINT	NOT NULL	FK → vehicles_vehiclecategory
<i>vehicle_id</i>	BIGINT	NULL	FK → vehicles_vehicle (assigned)
<i>driver_id</i>	BIGINT	NULL	FK → accounts_user (driver)
base_price	DECIMAL(10,2)	NOT NULL	Base price
extras_price	DECIMAL(10,2)	NOT NULL	Total extras cost
discount	DECIMAL(10,2)	NOT NULL	Discount amount
total_price	DECIMAL(10,2)	NOT NULL	Final total price
currency	VARCHAR(3)	NOT NULL	Currency code (MAD)
status	VARCHAR(20)	NOT NULL	PENDING, CONFIRMED, ASSIGNED, IN_PROGRESS, COMPLETED, CANCELLED
is_round_trip	BOOLEAN	NOT NULL	Round trip flag
return_datetime	TIMESTAMP TZ	NULL	Return pickup time
<i>return_transfer_id</i>	BIGINT	NULL	FK → transfers_transfer (self)

special_requests	TEXT	NOT NULL	Customer notes
internal_notes	TEXT	NOT NULL	Admin notes
created_at	TIMESTAMPTZ	NOT NULL	Booking creation
updated_at	TIMESTAMPTZ	NOT NULL	Last update

**Foreign Keys:** customer\_id → accounts\_user, vehicle\_category\_id → vehicles\_vehiclecategory, vehicle\_id → vehicles\_vehicle, driver\_id → accounts\_user, return\_transfer\_id → transfers\_transfer

**Indexes:** booking\_ref (UNIQUE), status, customer\_id, vehicle\_id, driver\_id

## locations\_zone

Geographic zones for zone-based pricing.

Column	Type	Nullable	Description
id	BIGINT	NOT NULL	Primary Key
name	VARCHAR(100)	NOT NULL	Zone name (default)
name_en	VARCHAR(100)	NULL	Name in English
name_fr	VARCHAR(100)	NULL	Name in French
name_ar	VARCHAR(100)	NULL	Name in Arabic
name_es	VARCHAR(100)	NULL	Name in Spanish
slug	VARCHAR(100)	NOT NULL	URL-friendly identifier (UNIQUE)
description	TEXT	NOT NULL	Zone description
center_latitude	DECIMAL(10,7)	NULL	Center point latitude
center_longitude	DECIMAL(10,7)	NULL	Center point longitude
radius_km	DECIMAL(8,2)	NULL	Zone radius in km
polygon_coordinates	JSONB	NULL	GeoJSON polygon (optional)
color	VARCHAR(7)	NOT NULL	Hex color for map display
is_active	BOOLEAN	NOT NULL	Active status
created_at	TIMESTAMPTZ	NOT NULL	Record creation
updated_at	TIMESTAMPTZ	NOT NULL	Last update

**Referenced by:** locations\_zonedistancerange, locations\_zonepricing

## locations\_route

Pre-defined routes between locations with fixed pricing.

Column	Type	Nullable	Description
id	BIGINT	NOT NULL	Primary Key
name	VARCHAR(200)	NOT NULL	Route name
slug	VARCHAR(200)	NOT NULL	URL-friendly identifier (UNIQUE)
description	TEXT	NOT NULL	Route description
origin_name	VARCHAR(200)	NOT NULL	Origin location name
origin_latitude	DECIMAL(10,7)	NULL	Origin GPS latitude
origin_longitude	DECIMAL(10,7)	NULL	Origin GPS longitude
origin_radius_km	DECIMAL(6,2)	NOT NULL	Origin pickup radius
destination_name	VARCHAR(200)	NOT NULL	Destination name
destination_latitude	DECIMAL(10,7)	NULL	Destination GPS latitude
destination_longitude	DECIMAL(10,7)	NULL	Destination GPS longitude
destination_radius_km	DECIMAL(6,2)	NOT NULL	Destination dropoff radius
distance_km	DECIMAL(8,2)	NOT NULL	Route distance
estimated_duration_minutes	INTEGER	NULL	Estimated travel time (≥0)
is_bidirectional	BOOLEAN	NOT NULL	Works both ways
is_popular	BOOLEAN	NOT NULL	Featured route flag
is_active	BOOLEAN	NOT NULL	Active status
order	SMALLINT	NOT NULL	Display order (≥0)
created_at	TIMESTAMPTZ	NOT NULL	Record creation
updated_at	TIMESTAMPTZ	NOT NULL	Last update

**Referenced by:** locations\_vehiclelroutepricing

## payments\_payment

Payment records with generic foreign key to booking types.

Column	Type	Nullable	Description
id	BIGINT	NOT NULL	Primary Key
payment_ref	VARCHAR(30)	NOT NULL	Unique payment reference (PAY-XXXX)
<a href="#">content_type_id</a>	INTEGER	NOT NULL	FK → django_content_type

object_id	INTEGER	NOT NULL	Related booking ID (≥0)
payment_type	VARCHAR(20)	NOT NULL	TRANSFER, TRIP, or RENTAL
<i>customer_id</i>	BIGINT	NULL	FK → accounts_user (optional)
customer_email	VARCHAR(254)	NOT NULL	Customer email
<i>gateway_id</i>	BIGINT	NOT NULL	FK → payments_paymentgateway
amount	DECIMAL(10,2)	NOT NULL	Payment amount
currency	VARCHAR(3)	NOT NULL	Currency code (MAD)
gateway_payment_id	VARCHAR(255)	NOT NULL	External payment ID
gateway_customer_id	VARCHAR(255)	NOT NULL	External customer ID
status	VARCHAR(20)	NOT NULL	PENDING, PROCESSING, COMPLETED, FAILED, REFUNDED
refunded_amount	DECIMAL(10,2)	NOT NULL	Total refunded
metadata	JSONB	NOT NULL	Additional data
error_message	TEXT	NOT NULL	Error details if failed
created_at	TIMESTAMPTZ	NOT NULL	Payment creation
updated_at	TIMESTAMPTZ	NOT NULL	Last update
completed_at	TIMESTAMPTZ	NULL	Completion timestamp

**Foreign Keys:** content\_type\_id → django\_content\_type, customer\_id → accounts\_user, gateway\_id → payments\_paymentgateway

**Indexes:** payment\_ref (UNIQUE), gateway\_payment\_id, (status, created\_at)

**Referenced by:** payments\_refund, payments\_invoice

## rentals\_rental

Car rental bookings with insurance and mileage tracking.

Column	Type	Nullable	Description
--------	------	----------	-------------

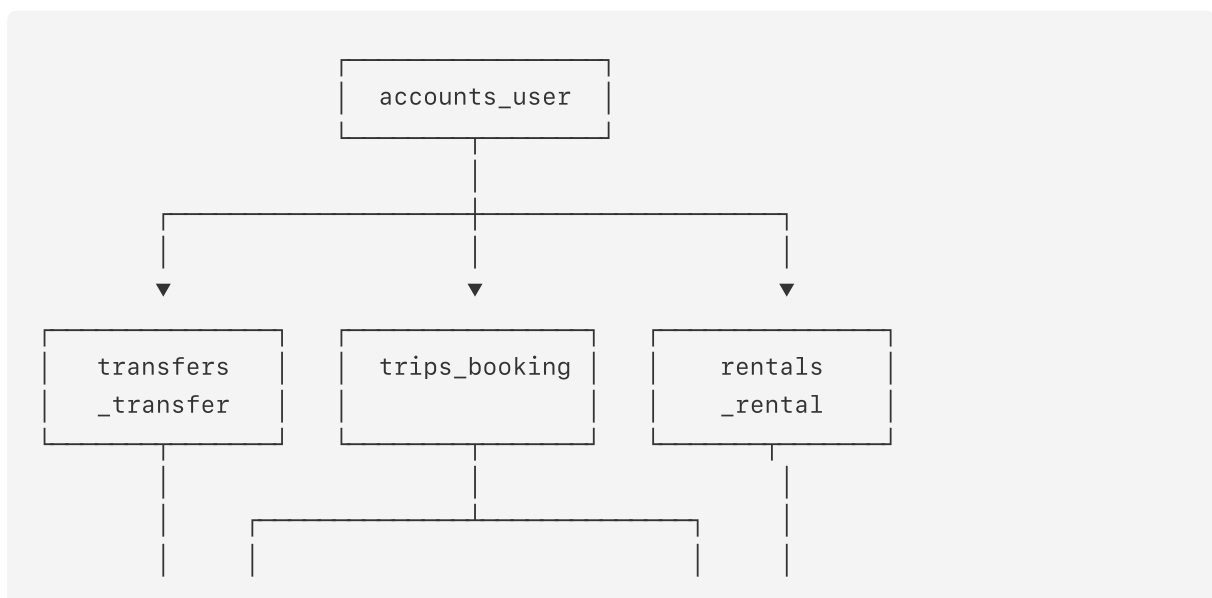
<b>id</b>	BIGINT	NOT NULL	Primary Key
booking_ref	VARCHAR(20)	NOT NULL	Unique booking reference (RNT-XXXX)
customer_name	VARCHAR(200)	NOT NULL	Customer full name
customer_email	VARCHAR(254)	NOT NULL	Customer email
customer_phone	VARCHAR(20)	NOT NULL	Customer phone
<i>customer_id</i>	BIGINT	NULL	FK → accounts_user (optional)
<i>vehicle_id</i>	BIGINT	NOT NULL	FK → vehicles_vehicle
driver_license_number	VARCHAR(50)	NOT NULL	License number
driver_license_expiry	DATE	NOT NULL	License expiry date
driver_date_of_birth	DATE	NOT NULL	Driver birth date
pickup_location	VARCHAR(200)	NOT NULL	Pickup location
return_location	VARCHAR(200)	NOT NULL	Return location
pickup_datetime	TIMESTAMPTZ	NOT NULL	Scheduled pickup
return_datetime	TIMESTAMPTZ	NOT NULL	Scheduled return
actual_pickup_datetime	TIMESTAMPTZ	NULL	Actual pickup time
actual_return_datetime	TIMESTAMPTZ	NULL	Actual return time
insurance_type	VARCHAR(20)	NOT NULL	NONE, BASIC, STANDARD, PREMIUM
daily_rate	DECIMAL(10,2)	NOT NULL	Daily rental rate
total_days	SMALLINT	NOT NULL	Rental duration (≥0)
subtotal	DECIMAL(10,2)	NOT NULL	Subtotal before extras
insurance_cost	DECIMAL(10,2)	NOT NULL	Insurance charge
extras_cost	DECIMAL(10,2)	NOT NULL	Total extras cost
different_location_fee	DECIMAL(10,2)	NOT NULL	Different return fee

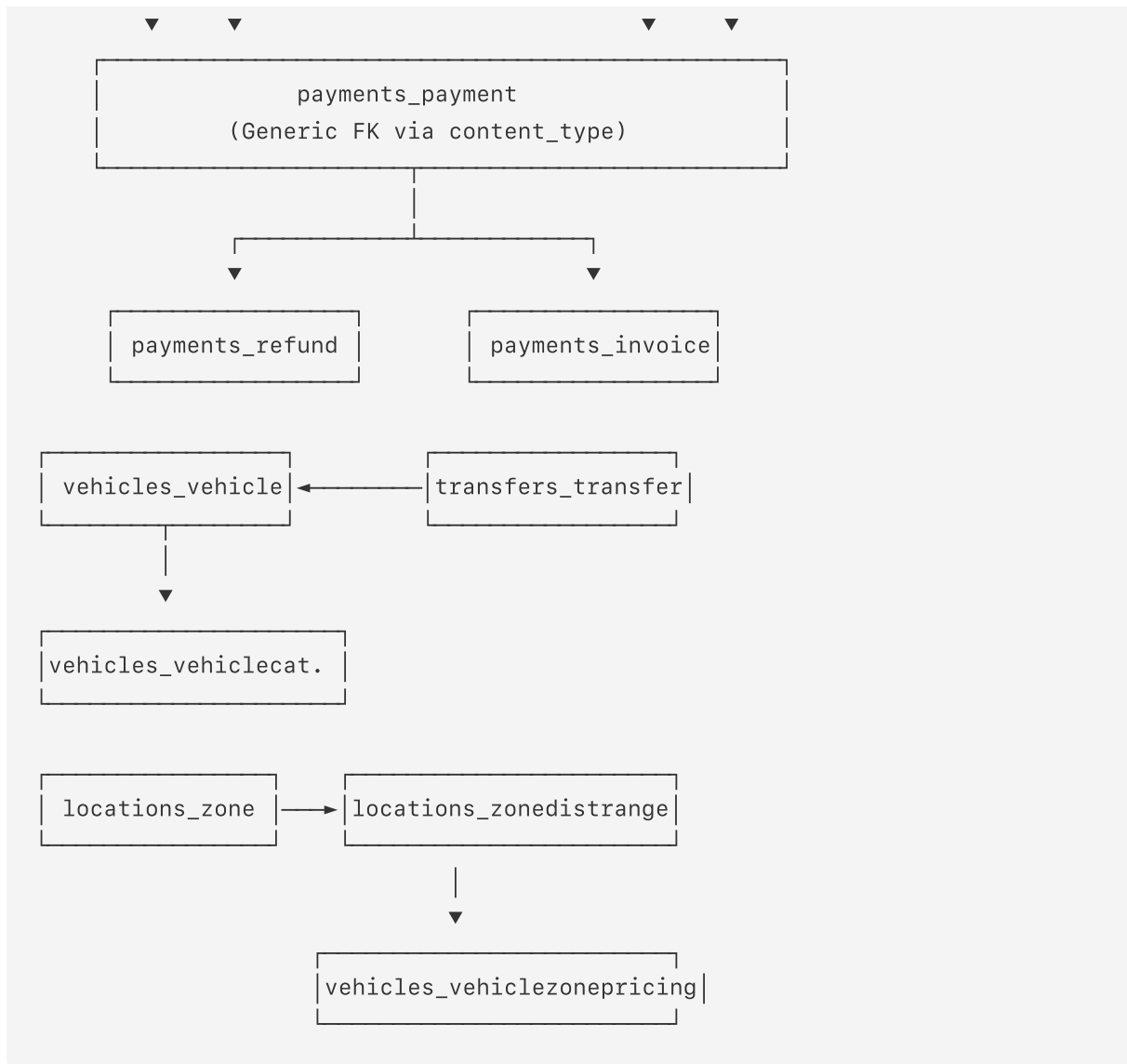
discount	DECIMAL(10,2)	NOT NULL	Discount amount
total_price	DECIMAL(10,2)	NOT NULL	Final total
deposit	DECIMAL(10,2)	NOT NULL	Security deposit
currency	VARCHAR(3)	NOT NULL	Currency code (MAD)
status	VARCHAR(20)	NOT NULL	PENDING, CONFIRMED, ACTIVE, COMPLETED, CANCELLED
pickup_mileage	INTEGER	NULL	Odometer at pickup (≥0)
return_mileage	INTEGER	NULL	Odometer at return (≥0)
fuel_level_pickup	VARCHAR(20)	NOT NULL	Fuel level at pickup
fuel_level_return	VARCHAR(20)	NOT NULL	Fuel level at return
special_requests	TEXT	NOT NULL	Customer notes
internal_notes	TEXT	NOT NULL	Admin notes
created_at	TIMESTAMPTZ	NOT NULL	Booking creation
updated_at	TIMESTAMPTZ	NOT NULL	Last update

**Foreign Keys:** customer\_id → accounts\_user, vehicle\_id → vehicles\_vehicle

**Indexes:** booking\_ref (UNIQUE), customer\_id, vehicle\_id

## Entity Relationship Diagram





## 7. API Endpoints

**Base URL:** `/api/v1/`

**Documentation:** `/api/docs/` (Swagger) | `/api/redoc/`

### Authentication (`/api/v1/auth/`)

POST <code>/register</code>	- Create account
POST <code>/token</code>	- Get JWT tokens
POST <code>/token/refresh</code>	- Refresh access token
GET/PUT <code>/profile</code>	- User profile
POST <code>/logout</code>	- Blacklist token

## Transfers (/api/v1/transfers/)

```
GET/POST /          - List/create transfers
POST /quote          - Price calculation
POST /{id}/assign-driver - Assign driver
GET /extras          - Available add-ons
```

## Locations (/api/v1/locations/)

```
GET /zones          - List zones
GET /routes          - List routes
GET /routes/popular - Popular routes
GET /calculate-distance - Distance calculation
```

## Vehicles (/api/v1/vehicles/)

```
GET /categories      - Vehicle categories
GET /features        - Available features
GET /                - List vehicles
```

## Payments (/api/v1/payments/)

```
GET /gateways        - Payment methods
POST /               - Initiate payment
POST /{id}/confirm   - Confirm payment
POST /{id}/refund    - Process refund
POST /webhooks/stripe - Stripe webhook
POST /webhooks/paypal - PayPal webhook
```

# 8. Authentication

---

## JWT Configuration

- Access token lifetime: 60 minutes
- Refresh token lifetime: 7 days
- Token rotation enabled on refresh
- Blacklisting enabled for logout



## Header Format

```
Authorization: Bearer <access_token>
```

## User Roles

Role	Permissions
ADMIN	Full access to all features
MANAGER	Manage bookings, vehicles, routes
DRIVER	View assigned transfers, update status
CLIENT	Create bookings, view own history

# 9. Payment Integration

## Supported Gateways

- **Stripe:** Payment intents, SCA/3D Secure, webhooks
- **PayPal:** Approval flow, redirect handling, webhooks
- **Bank Transfer:** Manual payment confirmation
- **Cash:** Pay on delivery/pickup

## Payment Flow

1. Client creates booking (transfer/trip/rental)
2. Client initiates payment (POST /api/v1/payments/)
3. Gateway returns client\_secret or approval\_url
4. Client completes payment on frontend
5. Client confirms payment (POST /{id}/confirm)
6. Webhook validates and finalizes payment
7. Invoice generated automatically

## Refund Processing

Full and partial refunds supported. Admin creates refund via API, gateway processes refund, status tracked with audit trail.

## 10. Dashboard Features

---

**URL:** `/dashboard/`

Staff-only admin interface built with Bootstrap 5.

### Available Features

- **Home:** Daily/monthly statistics, revenue charts, recent bookings
- **Transfers:** List, filter, search, detail view, driver assignment
- **Trips:** Package management, booking list, schedule management
- **Rentals:** Rental list, status tracking, mileage/fuel logging
- **Vehicles:** Fleet management, images, zone/route pricing
- **Zones:** Geographic zone configuration, distance ranges
- **Routes:** Origin-destination pairs, per-vehicle pricing
- **Users:** User management by role, search and filter
- **Payments:** Payment list, status filtering, refund management
- **Reports:** Revenue by type, top routes, booking analytics
- **Settings:** API keys, site configuration, payment gateways

## 11. Deployment

---

### Docker Compose Services

```
services:
  web:      Django + Gunicorn (port 8000)
  db:      PostgreSQL 16 (port 5432)
  redis:    Redis 7 (port 6379)
  celery:   Celery worker
  celery-beat: Celery scheduler
  nginx:    Reverse proxy (production)
```

### Environment Variables

```
SECRET_KEY, DEBUG, ALLOWED_HOSTS
DB_NAME, DB_USER, DB_PASSWORD, DB_HOST
REDIS_URL
```

```
STRIPE_PUBLIC_KEY, STRIPE_SECRET_KEY, STRIPE_WEBHOOK_SECRET  
PAYPAL_CLIENT_ID, PAYPAL_CLIENT_SECRET, PAYPAL_MODE  
GOOGLE_MAPS_API_KEY  
EMAIL_HOST, EMAIL_HOST_USER, EMAIL_HOST_PASSWORD
```

## Quick Start

```
# Copy environment file  
cp .env.example .env  
  
# Start all services  
docker-compose up -d  
  
# Run migrations  
docker-compose exec web python manage.py migrate  
  
# Create superuser  
docker-compose exec web python manage.py createsuperuser
```

## URLs

- **Dashboard:** <http://localhost:8000/dashboard/>
- **API Docs:** <http://localhost:8000/api/docs/>
- **Admin:** <http://localhost:8000/admin/>