# Telebat Software Requirements Specification (SRS)

**Introduction**

**Purpose**

This SRS document defines the functional and non-functional requirements for **Telebat**, a Laravel 12 API based multi-vendor marketplace with single-cart, single-store checkout. Telebat enables **Users**, **Vendors**, and **Admins** to interact through RESTful endpoints in a fully stateless manner.

**Scope**

Telebat provides:

* User-facing shopping API: account creation/verification, product browsing, cart, order and payment flows.
* Vendor-facing endpoints: store and product management, order fulfillment dashboard.
* Admin-facing capabilities: section management.
* Payment integration with Stripe and PayPal.
* Notification delivery, profile management, and locale-specific data support.

**Overall Description**

**Product Perspective**

Telebat is implemented as a stateless REST API. Each HTTP request carries all required context (authentication, locale). It integrates with external services (Stripe, PayPal, email/SMS platforms) and uses Laravel Sanctum for bearer token handling. No sessions are stored server-side, this supports horizontal scaling and simplifies backend architecture.

**User Classes**

* **User**: signs up, verifies via OTP/email, browses products, manages cart, places orders, receives notifications.
* **Vendor**: registers, verifies, creates stores/products, processes orders via dashboard.
* **Admin**: manages marketplace sections/categories, monitors system-wide configurations.

**Operating Environment**

* **Server**: Laravel 12 on PHP 8+, MySQL, HTTPS enforced.
* **API Client**: Compatible with web/mobile frontends; expects JSON responses and supports locale header for localization.

**Functional Requirements**

**1. Account & Authentication**

* Users, Vendors, Admins must **register** using email, phone, password, and role selection.
* Send OTP to phone and email; account becomes active after both verifications.
* Must support **login** with bearer token issuance and FCM token storage.
* Provide OTP resend, email verification, password reset with OTP flow (password/verify-otp, password/reset).
* Support password confirmation and logout endpoints.

**2. User Profile Management**

* Users may **view**, **update**, and **delete** their account via settings/profile.
* Users may **change password** securely via settings/password.

**3. Sections (Categories) – Admin Only**

* Users, Vendors and Admins may view marketplace sections, but only Admins can **create**, **update**, and **delete** marketplace sections, each with multilingual name, description, and image.

**4. Stores – Vendor Only**

* Users, Vendors and Admins may **view all stores**, but only Vendors may **add**, **update**, or **delete** their own stores.
* Store attributes include multilingual names, location coordinates, section, and image.

**5. Products – Vendor Management, User Browsing**

* Vendors may **add**, **update**, **delete** products linked to their stores.
* Products include multilingual titles and descriptions, price, stock, image.
* Users may **view** products with optional filters: keyword, section, store, price, quantity, sorting.

**6. Favorites – User Only**

* Users may **add** or **remove** products from favorites and **list** their favorites.

**7. Cart – User Only**

* Users may **view**, **add**, **update quantity**, **remove**, or **clear** the shopping cart.
* Cart enforces single-cart across all stores and cart products must belong to the same store.

**8. Orders – User Only**

* Users can:
  + Retrieve supported currencies.
  + View "My Orders" or a specific order.
  + Create new orders (checkout), update quantities, remove order items, or **cancel** orders.

**9. Payments – User Only**

* Users can initiate **Stripe payment intents** per order and **PayPal checkouts**.
* Stripe webhooks update payment status automatically.
* Users can **download invoices** via invoice endpoint.

**10. Vendor Dashboard – Vendor Only**

* Vendors may view their **stores list**, **store-specific orders**, and update order statuses to **delivering** or **delivered**.

**11. Notifications – User & Vendor**

* Both roles can **fetch notifications**, mark individual notifications as read, list unread, or mark all as read.

**12. Home Endpoint – User Only**

* Provides a personalized home dashboard (e.g., featured products, recommended stores) via /home.

**Non‑Functional Requirements**

| **Category** | **Requirement** |
| --- | --- |
| Performance | API responses must complete within 300 ms (95th percentile) |
| Security | Enforce HTTPS; use bearer tokens; sensitive data encrypted in transit and at rest |
| Localization | All content must support locale header (e.g., en, ar) |
| Scalability | Stateless design to support horizontal scaling under load |
| Maintainability | Validation rules and error formats must be consistent across endpoints |
| Availability | External integrations (payments, email/SMS) must have retry/fallback mechanisms |

**Constraints and Assumptions**

* Stateless design: all context via HTTP headers—strict REST principles.
* Dependencies include Stripe, PayPal, Laravel Sanctum, SMS/OTP providers.
* Assumes stable network and 3G+ mobile connections.
* No support for guest cart or guest checkout; users must register/login.

**Glossary**

* **Bearer Token**: Authentication token issued after login.
* **OTP**: One-Time Password for verification.
* **FCM Token**: Firebase Cloud Messaging token for push notifications.
* **Stateless API**: No session stored; each request stands alone.
* **Workspace**: Referent to a vendor’s physical store.