# Telebat Use Cases:

**Overview**

**Telebat** is a stateless, API-driven marketplace for **Users**, **Vendors**, and **Admins**. Use cases below map key interactions through REST endpoints.

**Actors**

* **End User** – browses, orders, pays, manages profile
* **Vendor** – manages store/products, fulfills orders
* **Admin** – manages sections.

**Use Case UC‑01: Register New Account**

| **Field** | **Description** |
| --- | --- |
| **Primary Actor** | End User / Vendor / Admin |
| **Goal** | Create verified account and obtain bearer token |
| **Preconditions** | Internet connection |
| **Trigger** | User submits registration form |
| **Main Flow** | 1. User calls POST /auth/register 2. System validates and creates account (unverified) 3. Sends OTP to email & phone 4. User verifies via OTP endpoints 5. System activates account and returns bearer token |
| **Alternate Flows** | OTP attempts exceeded → error message, Incorrect OTP → user retries, Email/phone already in use → validation error. |
| **Postconditions** | Account is active and authenticated |

**Use Case UC‑02: Login**

| **Field** | **Description** |
| --- | --- |
| **Primary Actor** | End User / Vendor / Admin |
| **Goal** | Authenticate and receive token |
| **Preconditions** | Account is verified |
| **Trigger** | Submit login credentials |
| **Main Flow** | 1. POST /auth/login with phone/email, password, fcm\_token 2. System validates, stores FCM, issues bearer token |
| **Alternate Flows** | Invalid credentials → 401 error, Account not verified → prompt reactivation |
| **Postconditions** | Bearer token issued; push notifications enabled |

**Use Case UC‑03: Manage Sections**

| **Field** | **Description** |
| --- | --- |
| **Primary Actor** | Admin |
| **Goal** | Create/update/delete marketplace sections |
| **Preconditions** | Logged in as Admin |
| **Trigger** | Admin accesses sections endpoints |
| **Main Flow** | 1. GET /sections 2. POST or PATCH section data (multilingual + image 3. System stores and confirms |
| **Alternate Flows** | Duplicate section error, Missing fields error |
| **Postconditions** | Sections updated correctly |

**Use Case UC‑04: Add New Store**

| **Field** | **Description** |
| --- | --- |
| **Primary Actor** | Vendor |
| **Goal** | Add a new store |
| **Preconditions** | Logged in as Vendor |
| **Trigger** | Submit store details |
| **Main Flow** | 1. POST /stores with name, section\_id, coordinates, image 2. System stores the store |
| **Alternate Flows** | Missing or invalid data → validation error, Image upload fails → retry prompt |
| **Postconditions** | New store exists for vendor |

**Use Case UC‑05: Manage Products**

| **Field** | **Description** |
| --- | --- |
| **Primary Actor** | Vendor |
| **Goal** | Add/edit/delete products |
| **Preconditions** | Vendor owns store |
| **Trigger** | Calls product endpoints |
| **Main Flow** | 1. POST or PATCH /products with necessary details 2. System saves product |
| **Alternate Flows** | Validation errors on input |
| **Postconditions** | Catalog updated successfully |

**Use Case UC‑06: Browse Products**

| **Field** | **Description** |
| --- | --- |
| **Primary Actor** | End User |
| **Goal** | Discover products |
| **Preconditions** | Authenticated |
| **Trigger** | GET /products with filters |
| **Main Flow** | 1. Client sends filter params (search, section, store, price range) 2. System returns matching products |
| **Alternate Flows** | No results → “no products found” response |
| **Postconditions** | User sees filtered products |

**Use Case UC‑07: Cart Management**

| **Field** | **Description** |
| --- | --- |
| **Primary Actor** | End User |
| **Goal** | Manage shopping cart |
| **Preconditions** | Authenticated |
| **Trigger** | Use cart endpoints |
| **Main Flow** | 1. POST to add, PUT to update quantity, DELETE to remove or clear, GET to view |
| **Alternate Flows** | Out-of-stock or invalid quantity → error |
| **Postconditions** | Cart reflects user's selections |

**Use Case UC‑08: Create Order (Checkout)**

| **Field** | **Description** |
| --- | --- |
| **Primary Actor** | End User |
| **Goal** | Finalize purchase |
| **Preconditions** | Cart has items |
| **Trigger** | POST /orders |
| **Main Flow** | 1. User posts order request 2. System creates order and returns currency options |
| **Alternate Flows** | Cart empty → error message |
| **Postconditions** | Order created and ready for payment |

**Use Case UC‑09: Payment Process**

| **Field** | **Description** |
| --- | --- |
| **Primary Actor** | End User |
| **Goal** | Complete order payment |
| **Preconditions** | Order exists |
| **Trigger** | Initiate payment via API |
| **Main Flow** | 1. POST to Stripe or PayPal endpoints 2. System returns payment intent or redirect link 3. Webhook updates status |
| **Alternate Flows** | Payment failure → return error |
| **Postconditions** | Payment status updated in system |

**Use Case UC‑10: Fulfill Orders**

| **Field** | **Description** |
| --- | --- |
| **Primary Actor** | Vendor |
| **Goal** | Mark order status |
| **Preconditions** | Order linked to vendor's store |
| **Trigger** | PATCH `/dashboard/orders/{order\_id}/delivering |
| **Main Flow** | 1. System updates status accordingly |
| **Alternate Flows** | Order not found or belongs to another vendor → error |
| **Postconditions** | Order status updated |

**Use Case UC‑11: Notifications**

| **Field** | **Description** |
| --- | --- |
| **Primary Actor** | End User / Vendor |
| **Goal** | Manage notifications |
| **Trigger** | Use notification endpoints |
| **Main Flow** | 1. GET unread/all /notifications 2. 2. PATCH individual or all read |
| **Alternate Flows** | Not applicable |
| **Postconditions** | Notifications updated |