**E‑Commerce Finance Module — BRSD v1**

Scope of this version: **Expenses**, **Incomes**, **Treasuries (Cash)**, and **Bank Accounts** — aligned with the UX shown in the screenshots (Daftra‑like forms).

**1) Business Goals & Non‑Goals**

**Goals**

* Record and control cash/bank movements via Expense & Income vouchers.
* Maintain master data for cash **Treasuries** and **Bank Accounts** with permissions.
* Post to a **double‑entry General Ledger** (GL) with tax, currency, and attachments.
* Support **draft → posted** lifecycle, approvals, and auditability.

**Non‑Goals (v1)**

* AP bills & AR invoices lifecycle (handled by Sales/Purchases modules).
* Checks/LCs & treasury forecasting.
* Multi‑entity consolidation.

**2) Roles & Permissions (RBAC)**

| **Role** | **Key Capabilities** |
| --- | --- |
| Finance Admin | Configure master data, charts, taxes, journals; post/reverse any voucher; view audit logs. |
| Cashier | Create/print Expense & Income vouchers tied to assigned treasuries/bank accounts; upload attachments. |
| Accountant | Review/approve/post vouchers; edit drafts; run reports. |
| Auditor (read‑only) | Read ledgers/vouchers/attachments & audit trail only. |

**Object‑level permissions**

* Treasury/Bank permissions: deposit\_permission, withdraw\_permission ∈ {Everyone, Role, User List}.
* Voucher actions: create, edit, post, reverse, delete (delete only drafts, never posted).

**3) Core Entities & Fields**

**3.1 ExpenseVoucher**

Represents money **leaving** a treasury or bank.

* id (UUID)
* code (string, seq; example: 000002)
* status ∈ {Draft, Posted, Reversed, Cancelled}
* date (date)
* amount (decimal, ≥ 0.00)
* currency\_code (FK Currency; default company currency e.g., EGP)
* fx\_rate (decimal; 1.0 for home currency)
* description (text)
* vendor\_id (FK Vendor; optional)
* category\_id (FK ExpenseCategory)
* source\_wallet\_type ∈ {Treasury, BankAccount}
* source\_wallet\_id (FK → Treasury or BankAccount)
* journal\_account\_id (FK GL Account) [default auto map]
* tax\_lines[] (see §3.5)
* attachments[] (files)
* recurrence\_id (nullable; link to schedule)
* Audit: created\_by, created\_at, posted\_by, posted\_at, reversed\_by, reversed\_at, revision.

**3.2 IncomeVoucher**

Represents money **entering** a treasury or bank.

Fields mirror ExpenseVoucher except customer\_id (optional) may replace vendor; keep generic counterparty\_id with counterparty\_type ∈ {Vendor, Customer, Other}.

**3.3 Treasury (Cash Safe)**

* id (UUID)
* name (string, unique per company)
* status ∈ {Active, Inactive}
* currency\_code
* description
* deposit\_permission (ACL)
* withdraw\_permission (ACL)
* balance (derived from ledger)

**3.4 BankAccount**

* id (UUID)
* bank\_name (string)
* name (string label visible to users)
* account\_number (string, masked in UI)
* currency\_code
* status
* description
* ACL: deposit/withdraw permissions (same as Treasury)
* balance (derived)

**3.5 TaxLine**

* tax\_id (FK Tax)
* base\_amount
* tax\_amount
* is\_withholding (bool)
* direction ∈ {Expense, Income}

**3.6 Shared Lookups**

* ExpenseCategory / IncomeCategory → FK to GL expense/revenue accounts.
* Currency (ISO code, symbol, decimals).

**4) Numbering & Lifecycle**

**Numbering**

* Per document type & fiscal year: EXP-YYYY-#### and INC-YYYY-#### (prefix configurable).
* Allows manual override with uniqueness check while Draft.

**Lifecycle**

1. **Draft**: editable; no ledger impact.
2. **Posted**: immutable financial fields; generates GL entries & updates cash/bank balances.
3. **Reversed**: create a system reversal voucher with opposite entries; original stays posted.

**Approvals (optional setting)**

* Workflow states: Draft → *Approved* → Posted.
* Approval rules per amount thresholds & role.

**5) Posting Rules (Double‑Entry)**

**5.1 Expense Voucher**

* **Debit:** Expense (from Category’s GL) = net amount (excl. VAT)
* **Debit:** Input VAT (if recoverable) = tax amount
* **Credit:** Cash/Bank (from source wallet control account) = gross amount
* **Optional:** Withholding tax → Credit: Tax Payable; Debit: Expense (or separate WHT receivable as per locale).

**5.2 Income Voucher**

* **Debit:** Cash/Bank (wallet control) = gross amount
* **Credit:** Revenue (from Category’s GL) = net amount
* **Credit:** Output VAT (tax payable) = tax amount
* **Optional:** Withholding tax withheld by customer → Debit: WHT Receivable; Credit: Cash/Bank.

**Exchange**

* If currency ≠ base, amounts stored in both txn\_currency and base\_currency via fx\_rate at posting time.
* FX differences handled at settlement/revaluation (out of v1 scope but model stores).

**Attachments**: stored in object storage, signed URLs.

**6) Validations & Business Rules**

* Amount > 0 for **Posted**.
* Date within open fiscal period.
* Source wallet active and user has permission (deposit/withdraw) matching voucher **direction**.
* Category must map to a leaf GL account of correct type (Expense/Revenue).
* Taxes must be configured for the chosen category/country; recompute on currency change.
* Prohibit delete when status != Draft. Use reverse instead.
* Enforce unique code per type+year.

**7) UI Specification (aligned to screenshots)**

**7.1 Expense/Add Form**

* **Amount** (numeric, currency selector)
* **Description** (multiline)
* **Attachments** (drag‑drop, preview)
* **Code Number** (auto, editable until posted)
* **Date** (default today)
* **Vendor** (autocomplete)
* **Category** (dropdown, searchable)
* **Journal Account** (auto; overrides allowed if permitted)
* **Source Wallet**: radio **Treasury/Bank** then picker ("Select Supplier" becomes wallet picker)
* **Tax**: add multiple tax lines
* **Recurring**: toggle opens schedule editor (see §9)
* **Buttons**: Save as Draft, Post/Save, Cancel.

**7.2 Income/Add Form**

Same layout; Vendor label switches to Counterparty (Customer/Vendor/Other).  
Journal Account auto to revenue.

**7.3 Treasury Add**

* Fields: Type fixed to **Treasury**, Status, Name\*, Description, Permissions (Deposit/Withdraw default Everyone).

**7.4 Bank Account Add**

* Fields: Bank Name\*, Name\*, Account Number\*, Currency, Status, Description, Permissions.

**Accessibility**: keyboard friendly, validation hints under fields; Arabic/English labels.

**8) Data Model (logical)**

Company(1) ──< Treasury >── GLAccount (cash)

└──< BankAccount >── GLAccount (bank)

└──< ExpenseVoucher >──< TaxLine >── Tax

└──< IncomeVoucher >──< TaxLine >── Tax

ExpenseVoucher ── category\_id ──> GLAccount(type=Expense)

IncomeVoucher ── category\_id ──> GLAccount(type=Revenue)

Voucher ──< LedgerEntry >── GLAccount (double‑entry)

**9) Recurring Vouchers**

* Schedules: daily/weekly/monthly/quarterly/custom RRULE; next occurrence creates a **Draft** with cloned fields and new code.
* Pause/Resume per user.

**10) API Contracts (REST)**

**10.1 Expense**

POST /api/finance/expenses

{

"code": "EXP-2025-0002",

"date": "2025-08-28",

"currency": "EGP",

"fx\_rate": 1,

"amount": 1150.00,

"description": "Office chairs",

"counterparty": {"type": "Vendor", "id": "vnd\_123"},

"category\_id": "gl\_7001",

"source": {"type": "Treasury", "id": "tre\_001"},

"tax\_lines": [{"tax\_id": "vat\_14", "base\_amount": 1000.0, "tax\_amount": 150.0, "direction": "Expense"}],

"attachments": ["file\_abc"]

}

POST /api/finance/expenses/{id}/post → posts & returns ledger lines.  
POST /api/finance/expenses/{id}/reverse → auto reversal.

**10.2 Income**

POST /api/finance/incomes (mirrors Expense; direction = Income).

**10.3 Master Data**

POST /api/finance/treasuries, POST /api/finance/bank-accounts with ACL fields.  
GET /api/finance/wallets?type=bank|treasury for wallet pickers.

**10.4 Ledger Preview**

POST /api/finance/vouchers/preview-ledger → returns would‑be entries without posting.

**11) Database Schema (DDL sketch)**

-- GL Accounts (simplified)

create table gl\_accounts(

id uuid primary key, code text unique, name text,

type text check (type in ('Asset','Liability','Equity','Revenue','Expense')),

is\_leaf boolean default true

);

create table treasuries(

id uuid primary key, company\_id uuid, name text unique,

status text check (status in ('Active','Inactive')),

currency\_code char(3) not null default 'EGP',

description text,

deposit\_acl jsonb, withdraw\_acl jsonb

);

create table bank\_accounts(

id uuid primary key, company\_id uuid,

bank\_name text, name text, account\_number text,

currency\_code char(3) not null default 'EGP',

status text check (status in ('Active','Inactive')),

description text,

deposit\_acl jsonb, withdraw\_acl jsonb

);

create table vouchers(

id uuid primary key, type text check (type in ('Expense','Income')),

code text, status text check (status in ('Draft','Posted','Reversed','Cancelled')),

date date, currency\_code char(3), fx\_rate numeric(18,6), amount numeric(18,2),

description text, category\_account\_id uuid references gl\_accounts(id),

wallet\_type text check (wallet\_type in ('Treasury','BankAccount')),

wallet\_id uuid, counterparty\_type text, counterparty\_id uuid,

journal\_account\_id uuid references gl\_accounts(id), recurrence\_id uuid,

created\_by uuid, created\_at timestamptz default now(), posted\_by uuid, posted\_at timestamptz

);

create table voucher\_taxes(

id uuid primary key, voucher\_id uuid references vouchers(id) on delete cascade,

tax\_id uuid, base\_amount numeric(18,2), tax\_amount numeric(18,2),

is\_withholding boolean default false, direction text

);

create table ledger\_entries(

id uuid primary key, voucher\_id uuid references vouchers(id),

gl\_account\_id uuid references gl\_accounts(id),

debit\_base numeric(18,2) default 0, credit\_base numeric(18,2) default 0,

debit\_txn numeric(18,2) default 0, credit\_txn numeric(18,2) default 0,

currency\_code char(3), fx\_rate numeric(18,6), entry\_date date

);

**12) Audit, Logs & Compliance**

* **Audit trail** per voucher: all edits, who/when, before/after.
* File retention policy & checksum for attachments.
* Egyptian e‑invoicing: *not part of vouchers* but tax configs align (keep Tax ID codes for mapping later).

**13) Reports (v1)**

* Cashbook by Treasury / Bank (daily running balance).
* Expense by Category, Vendor, Date range.
* Income by Category, Counterparty, Date range.
* Tax summary (Input vs Output VAT); WHT ledger.

**14) Test Scenarios (Happy & Edge)**

1. Create Expense (EGP 1,000 + 14% VAT) from Treasury A; post → GL matches rules; Treasury balance decreases 1,140.
2. Create Income (EGP 5,000 no VAT) to Bank B; post → Bank balance increases 5,000.
3. Draft deletion allowed; posted deletion blocked; reverse creates opposite entries.
4. User without Withdraw permission for Treasury A cannot post Expense from it.
5. Multi‑currency Income (USD) with fx\_rate recorded; ledger stores both txn/base.
6. Withholding tax on Income reduces cash and creates WHT receivable.

**15) Open Questions / Next Decisions**

* Do we allow negative lines (refunds) or separate Refund voucher type?
* Require category per line (multi‑line voucher) vs single category per voucher (screens show single; v1 keeps single).
* Approval thresholds & SLAs.
* Printing templates (Arabic/English).

**16) Delivery Plan (slice by slice)**

1. **Master Data** (Treasuries, Bank Accounts + ACL) + wallet picker API.
2. **Expense/Income Draft CRUD** + tax calc + attachments.
3. **Posting Engine** + ledger tables + reports: Cashbook & Expense by Category.
4. **Reversal & Audit** + recurring scheduler.

End of BRSD v1 for the Finance Module (Expenses, Incomes, Treasuries & Bank Accounts).