

# UCB Stock CRM and Risk Management Platform

Developer Instructions v1.0 | Date: February 12, 2026 | Prepared by: UCB Stock Management

## 1. Project Overview and Scope

Build an internal CRM and Risk Management Platform for UCB Stock brokerage. The system must:

1. Ingest daily data: admin balances (Excel), DSE trade files (XML), CSE trade files (Text), and deposit/withdrawal transactions (Excel).
2. Maintain real-time client positions, cash balances, and margin accounts.
3. Enforce BSEC Margin Rules 2025 (gazetted November 6, 2025) and portfolio concentration limits.
4. Generate client statements, risk reports, margin alerts, and contract notes.
5. Provide a web dashboard for risk managers and relationship managers.

IN SCOPE: Data ingestion, trade processing, holdings management, cash ledger, margin engine, risk alerts, reporting, client portfolio dashboard, user roles.

OUT OF SCOPE (for now): Mobile app, real-time price streaming, machine learning, client-facing portal, OMS integration.

## 2. Data Sources and File Formats

### 2.1 Opening Balances (Admin Balance) - Excel

Source: Excel file containing client-wise positions and cash as of 31-Jan-2026.

Frequency: One-time import as baseline, then updated at each month-end or as needed.

Expected columns (confirm with actual file):

- BOID / ClientCode: unique client identifier
- SecurityCode / ISIN: for each security held
- Quantity: number of shares held
- AveragePrice: weighted average purchase cost
- CashBalance: available cash in the account
- LoanBalance: outstanding margin loan (if any)

- ClientName: full name
- ClientCategory: retail, institution, foreign, etc.

Process:

- Validate all rows. Reject rows with missing BOID or negative quantities.
- Import into clients, holdings, cash\_accounts tables.
- Set initial average\_cost and total\_invested from provided cost data.
- Log import in import\_audit table with filename, row counts, timestamp.

## 2.2 DSE Trade File - XML

Source: Daily XML file from DSE (e.g., 20260201-144801-trades-UBR-out.xml).

Frequency: Daily, after market close.

Size: Can be 30MB+, containing hundreds of thousands of rows.

Key XML Attributes:

- Action: NEW, EXEC, RPLREQ, CXLREQ (order lifecycle)
- Status: ACK, RPLD, PF, FILL, CXLD, EXPIRED, REJ
- OrderID: unique order identifier
- RefOrderID: references previous order (for replacements/cancellations)
- Side: B (Buy), S (Sell)
- BOID / ClientCode: client identifier (map both to single client master)
- SecurityCode / ISIN: security identifier (use ISIN as unique key)
- Board: PUBLIC, BLOCK, SPUBLIC
- Date, Time: trade date and time
- Quantity: executed quantity (for fills)
- Price: execution price
- Value: Quantity x Price (in BDT)
- ExecID: unique execution ID (CRITICAL for de-duplication)
- Session: PREOPEN, CONTINUOUS, POSTCLOSING
- FillType: PF (partial fill), FILL (full fill), WON (block trade)
- Category: A, B, Z, S (determines margin eligibility)
- AssetClass: EQ (equity), MF (mutual fund)
- CompulsorySpot: flag for spot market trades
- TraderDealerID, OwnerDealerID: dealer identifiers

## 2.3 CSE Trade File - Text

Source: Daily text file from CSE (e.g., BT\_WITH\_TRADE\_FLAG.txt).

Frequency: Daily, after market close.

Format: Pipe-delimited or fixed-width text (developer must confirm delimiter from sample file).

ACTION REQUIRED: Developer must parse the provided sample CSE file and document the exact field mapping. Expected fields should include: ClientCode, SecurityCode, ISIN, Side, Quantity, Price, Value, TradeDate, TradeTime, ExecID, Category.

The CSE file must be normalized to the same internal structure as DSE trades before processing.

## 2.4 Deposit/Withdrawal Transactions - Excel

Source: Excel file (e.g., Deposit Withdrawal 01.02.2026.xlsx).

Frequency: Daily.

Expected columns:

- BOID / ClientCode
- TransactionDate
- Amount (positive for deposit, negative for withdrawal)
- Type: DEPOSIT, WITHDRAWAL, CHEQUE\_DEPOSIT, BANK\_TRANSFER, etc.
- Reference: cheque number, transfer ID, etc.
- Narration: description

Process:

- Validate: withdrawal cannot exceed available cash + margin headroom.
- Update client cash balance via cash\_ledger.
- If withdrawal would trigger margin shortfall, flag and reject.

## 3. Database Schema

Use PostgreSQL (Supabase). All tables must have created\_at and updated\_at timestamps. Enable Row Level Security where appropriate.

### 3.1 clients

- client\_id (PK, UUID)
- bo\_id (unique, text) -- beneficiary owner ID from CDBL
- client\_code (unique, text) -- short broker code
- name (text)
- category (text) -- retail, institution, foreign
- income\_status (text) -- employed, self\_employed, student, homemaker, retired
- is\_margin\_eligible (boolean, default false)
- margin\_eligibility\_date (date) -- when they became eligible

- rm\_id (FK to users) -- assigned relationship manager
- phone, email, address (text)
- status (text) -- active, suspended, closed
- kyc\_completed (boolean)
- created\_at, updated\_at

## 3.2 securities

- isin (PK, text)
- security\_code (unique, text) -- e.g. RENATA, BRACBANK
- company\_name (text)
- asset\_class (text) -- EQ, MF, BOND, GOVT
- category (text) -- A, B, Z, N, G, S (updated by admin)
- board (text) -- PUBLIC, BLOCK, SPUBLIC, SME, ATB
- lot\_size (int, default 1)
- face\_value (numeric)
- sector (text) -- e.g. Banking, Pharma, Engineering
- free\_float\_market\_cap (numeric) -- in BDT, for margin eligibility check
- trailing\_pe (numeric) -- for margin eligibility check
- last\_close\_price (numeric)
- is\_marginable (boolean, default false) -- computed from rules
- margin\_rate (numeric) -- override per security if needed
- status (text) -- active, suspended
- created\_at, updated\_at

## 3.3 raw\_trades (staging table)

- id (PK, bigserial)
- source (text) -- DSE, CSE
- file\_name (text)
- action, status, order\_id, ref\_order\_id, side, bo\_id, client\_code
- isin, security\_code, board, trade\_date, trade\_time
- quantity (int), price (numeric), value (numeric)
- exec\_id (text), session, fill\_type, category, asset\_class
- compulsory\_spot (boolean)
- raw\_data (jsonb) -- store entire original row
- processed (boolean, default false)
- created\_at

### 3.4 trade\_executions

- exec\_id (PK, text) -- from ExecID, ensures no duplicates
- order\_id (text)
- client\_id (FK to clients)
- isin (FK to securities)
- exchange (text) -- DSE, CSE
- side (text) -- BUY, SELL
- quantity (int)
- price (numeric)
- value (numeric)
- trade\_date (date)
- trade\_time (time)
- settlement\_date (date) -- computed: trade\_date + settlement cycle
- session (text)
- fill\_type (text)
- category (text) -- snapshot of security category at trade time
- board (text)
- commission (numeric) -- computed from fee schedule
- exchange\_fee (numeric)
- cdbl\_fee (numeric)
- ait (numeric) -- advance income tax
- net\_value (numeric) -- value +/- all fees
- created\_at

### 3.5 holdings

- client\_id (FK, composite PK with isin)
- isin (FK, composite PK with client\_id)
- quantity (int) -- net shares held
- average\_cost (numeric) -- weighted average cost per share (includes commission)
- total\_invested (numeric) -- cumulative cost of all buys
- realized\_pl (numeric) -- cumulative realized profit/loss from sells
- as\_of\_date (date)
- updated\_at

Update logic:

- BUY:  $\text{new\_avg} = (\text{old\_qty} * \text{old\_avg} + \text{buy\_value\_with\_fees}) / (\text{old\_qty} + \text{buy\_qty})$
- SELL:  $\text{realized\_pl} += (\text{sell\_net\_value} - \text{average\_cost} * \text{sell\_qty})$
- SELL:  $\text{quantity} -= \text{sell\_qty}$  (average\_cost does NOT change on sells)

### 3.6 cash\_ledger

Use a LEDGER model, not a single balance row. Every transaction creates a new ledger entry. Balance is computed as running sum.

- id (PK, bigserial)
- client\_id (FK)
- transaction\_date (date)
- value\_date (date) -- settlement date for trades
- amount (numeric) -- positive = credit, negative = debit
- running\_balance (numeric) -- computed
- type (text) -- DEPOSIT, WITHDRAWAL, BUY\_TRADE, SELL\_TRADE, COMMISSION, TAX, DIVIDEND, IPO\_ALLOTMENT, INTEREST\_CHARGE
- reference (text) -- ExecID for trades, cheque no. for deposits, etc.
- narration (text)
- created\_at

Available Cash = running\_balance of latest ledger entry for the client.

Settled Cash = sum of entries where value\_date <= today.

Unsettled Cash = sum of entries where value\_date > today.

### 3.7 margin\_accounts

- client\_id (PK, FK)
- loan\_balance (numeric)
- margin\_ratio (numeric) -- current equity:loan ratio
- portfolio\_value (numeric) -- market value of all marginable holdings
- client\_equity (numeric) -- portfolio\_value - loan\_balance
- maintenance\_status (text) -- NORMAL, WARNING, MARGIN\_CALL, FORCE\_SELL
- last\_margin\_call\_date (date)
- margin\_call\_count (int) -- consecutive unresolved calls
- updated\_at

### 3.8 margin\_alerts

- id (PK, bigserial)
- client\_id (FK)
- alert\_date (date)
- alert\_type (text) -- MARGIN\_CALL, FORCE\_SELL\_TRIGGERED, CONCENTRATION\_BREACH, CATEGORY\_CHANGE
- details (jsonb) -- required\_equity, available\_equity, shortfall, etc.
- notification\_sent (boolean)

- notification\_channels (text[]) -- email, sms, whatsapp
- resolved (boolean, default false)
- resolved\_date (date)
- resolved\_by (FK to users)
- created\_at

### 3.9 daily\_prices

- isin (FK, composite PK with date)
- date (date, composite PK with isin)
- open\_price (numeric)
- high\_price (numeric)
- low\_price (numeric)
- close\_price (numeric)
- volume (bigint)
- value (numeric)
- num\_trades (int)
- source (text) -- DSE, CSE

Note: If a security did not trade on a given day, carry forward the last available close\_price.

### 3.10 daily\_snapshots

Capture end-of-day state for each client for historical reporting.

- client\_id (FK, composite PK with snapshot\_date)
- snapshot\_date (date)
- total\_portfolio\_value (numeric)
- cash\_balance (numeric)
- loan\_balance (numeric)
- net\_equity (numeric)
- margin\_utilization\_pct (numeric)
- unrealized\_pl (numeric)
- created\_at

### 3.11 fee\_schedule (configurable)

- id (PK)
- fee\_type (text) -- BROKERAGE\_COMMISSION, EXCHANGE\_FEE, CDBL\_FEE, AIT, LAGA
- rate (numeric) -- percentage or fixed amount
- min\_amount (numeric)

- max\_amount (numeric)
- applies\_to (text) -- BUY, SELL, BOTH
- effective\_from (date)
- effective\_to (date, nullable)
- is\_active (boolean)

See Section 7 for current fee rates.

### 3.12 import\_audit

- id (PK)
- file\_name (text)
- file\_type (text) -- ADMIN\_BALANCE, DSE\_TRADE, CSE\_TRADE, DEPOSIT\_WITHDRAWAL, PRICE\_DATA
- import\_date (timestamp)
- total\_rows (int)
- processed\_rows (int)
- rejected\_rows (int)
- error\_details (jsonb)
- status (text) -- SUCCESS, PARTIAL, FAILED
- imported\_by (FK to users)

## 4. Data Ingestion Pipeline

Step 1 - File Upload: Admin uploads files through web dashboard (or files are auto-detected from a watched folder/Supabase Storage bucket).

Step 2 - Parse and Stage: Parse XML (DSE), Text (CSE), or Excel files. Insert ALL rows into raw\_trades staging table. Log import in import\_audit.

Step 3 - Filter and Validate:

- Filter raw\_trades to only Status IN ('FILL', 'PF') AND Quantity > 0.
- Deduplicate on exec\_id. If exec\_id already exists in trade\_executions, skip.
- Validate: client must exist in clients table. If unknown BOID, create a placeholder client record and flag for review.
- Validate: security must exist in securities table. If unknown ISIN, create placeholder and flag.

Step 4 - Compute Fees: For each valid trade execution, calculate commission, exchange fees, CDBL fees, AIT per the fee\_schedule table. Compute net\_value.

Step 5 - Compute Settlement Date:



- Category A, B, G, N:  $\text{settlement\_date} = \text{trade\_date} + 2 \text{ business days}$
- Category Z:  $\text{settlement\_date} = \text{trade\_date} + 3 \text{ business days (buy)}$
- Spot trades (CompulsorySpot = true): T+0 for sell, T+1 for buy

Step 6 - Update Holdings: Apply buy/sell logic to holdings table per the formulas in Section 3.5.

Step 7 - Update Cash Ledger: Create ledger entries for each trade:

- BUY: debit (negative) for trade value + fees
- SELL: credit (positive) for trade value - fees
- Set  $\text{value\_date} = \text{settlement\_date}$

Step 8 - Recalculate Margin: After processing all trades for the day, trigger margin recalculation for all affected clients (see Section 6).

Step 9 - Generate Alerts: If any client breaches margin thresholds, create `margin_alerts` records and send notifications.

Step 10 - EOD Snapshot: At end of day, create `daily_snapshots` for all active clients.

Error Handling:

- If any step fails for a trade, mark it as failed in `raw_trades` with error details.
- Continue processing remaining trades (do not abort entire batch).
- Provide a summary report of failed records for manual review.
- Support re-processing of failed records after manual correction.

## 5. Business Rules - Trade Processing

ONLY records with Status = FILL or PF (partial fill) AND Quantity > 0 change positions and cash. All other statuses (ACK, RPLD, CXLD, REJ, EXPIRED) are informational and stored in `raw_trades` for audit only.

Order Lifecycle:

NEW (ACK) -> may be replaced via RPLREQ (RPLD) -> new order -> eventually PF/FILL or CXLD/EXPIRED.

Replacements (Action=RPLREQ, Status=RPLD): The old order is cancelled and a new order is created. Track chain via RefOrderID. Only final executed fills matter for positions.

Cancellations (Action=CXLREQ, Status=CXLD): Order cancelled. Any fills that occurred before cancellation remain valid.

Duplicate Prevention: ExecID is the unique key. NEVER process the same ExecID twice. Use UPSERT or check-before-insert.

Multi-Exchange Reconciliation:

- Same security may trade on both DSE and CSE.
- Use ISIN as the canonical security identifier to consolidate.
- Holdings are consolidated per client per ISIN regardless of exchange.
- Track exchange in trade\_executions for reporting purposes.

## 6. Business Rules - Margin and Risk (BSEC Margin Rules 2025)

Reference: BSEC (Margin) Rules, 2025 - Gazette dated November 6, 2025. This section MUST be implemented exactly per the gazetted rules.

### 6.1 Eligible Securities for Margin Financing

ONLY the following securities are eligible for margin loans:

- Category A shares listed on Main Board
- Category B shares listed on Main Board, BUT ONLY if the company pays minimum 5% annual dividend
- Must have Free Float Market Cap of at least BDT 50 Crore
- Must have Trailing P/E ratio not exceeding 30 (or double the sectoral median P/E, whichever is lower)

NOT eligible:

- Category Z shares
- Securities on SME, ATB, or OTC platforms
- Locked-in shares, lien shares, or directors shares
- Securities of Life Insurance companies without up-to-date actuarial valuation

The is\_marginable flag on the securities table must be recomputed daily based on these rules. Admin can also manually override.

Category Change Rule: If a margin-financed A or B category share is later converted to Z category, or if a B category company fails to pay 5% dividend, the system must:

1. Flag the affected client accounts
2. Notify the client
3. Track a 60-trading-day deadline for compulsory sell and adjustment

## 6.2 Client Eligibility for Margin

Clients must meet ALL of the following to be eligible:

- Must have had average minimum investment of BDT 5 Lakh over the preceding year
- Must have stable income (NOT students, homemakers, or retired persons, unless homemakers/retirees qualify as High Net-Worth Individual)
- Must have completed KYC
- Margin agreement valid for 1 year, requires mutual consent for renewal
- Margin can only be provided to individual investors (not joint or cash basis)

The system must check client eligibility before allowing any margin facility. Store eligibility status in `clients.is_margin_eligible`.

Conflict of Interest: Margin cannot be extended to the financier's own directors, employees, or their family members. Track this in client records.

## 6.3 Margin Financing Ratios

Margin ratios are based on portfolio size and market conditions:

Portfolio Value BDT 5 Lakh to less than BDT 10 Lakh: Max ratio 1:0.5 (client puts 1, can borrow 0.5)

Portfolio Value BDT 10 Lakh or more: Max ratio 1:1 (client puts 1, can borrow 1)

Override condition: If overall market P/E exceeds 20, ALL margin ratios are capped at 1:0.5 regardless of portfolio size.

Life Insurance companies: Margin capped at 1:0.25 (must have current actuarial valuation).

Margin loans can ONLY be used to purchase shares. No cash withdrawals or transfers from margin are permitted.

Unrealized gains CANNOT be used to access margin loans. Only realized equity counts.

Capital Limit: Total margin financing by UCB Stock cannot exceed 3x core capital/net worth.

## 6.4 Margin Maintenance and Thresholds

CRITICAL - Three threshold levels must be monitored continuously:

Level 1 - NORMAL: Client equity  $\geq$  75% of margin finance, OR portfolio value  $\geq$  175% of margin finance. No action needed.

Level 2 - MARGIN CALL: Client equity drops below 75% of margin finance (portfolio value < 175% of margin finance). System must:

- Immediately generate MARGIN\_CALL alert
- Notify client via writing, email, SMS, and WhatsApp
- Record in margin\_alerts table
- Track consecutive unresolved calls
- If client fails to meet margin call 3 consecutive times, broker may sell with 7-day notice

Level 3 - FORCE SELL: Client equity drops to 50% or less of margin finance (portfolio value <= 150% of margin finance). System must:

- Immediately generate FORCE\_SELL\_TRIGGERED alert
- NO prior notice required - broker is OBLIGATED to sell immediately
- Auto-generate force sell report for operations team
- Calculate minimum shares to sell to restore margin to Level 1

Formulas:

- Portfolio Value = SUM(quantity \* last\_close\_price) for all marginable holdings
- Client Equity = Portfolio Value - Loan Balance
- Maintenance Ratio = Client Equity / Loan Balance (must be >= 0.75)
- Portfolio Ratio = Portfolio Value / Loan Balance (must be >= 1.75)

Quarterly Interest: Margin interest must be paid quarterly. Track due dates and flag overdue accounts.

## 7. Business Rules - Fees and Commissions

Every trade must have fees calculated and deducted. Store all fee rates in the fee\_schedule table so they can be updated without code changes.

Current Fee Structure (as of February 2026):

1. Brokerage Commission: Negotiable, max 1% of trade value. UCB Stock typically charges [INSERT YOUR RATE, e.g., 0.30%] on both buy and sell.
2. DSE/CSE Exchange Fee (Laga): 0.03% of trade value (turnover).
3. CDBL Transaction Fee: 0.0175% of trade value (market value of shares), minimum BDT 5 per transaction.
4. BSEC Fee: Included in exchange fee structure.

5. Advance Income Tax (AIT): 0.05% of trade value, collected at source by the exchange on both buy and sell transactions.

6. Capital Gains Tax: 15% on capital gains from listed securities (for companies, funds, trusts). For individual investors, gains within 5 years are added to total income; gains after 5 years are taxed at 15%.

For each trade, compute:

- BUY total cost = trade\_value + commission + exchange\_fee + cdbl\_fee + ait
- SELL net proceeds = trade\_value - commission - exchange\_fee - cdbl\_fee - ait

Store individual fee components in trade\_executions for transparency and contract note generation.

IMPORTANT: Include commission in the average cost calculation for holdings.

## 8. Business Rules - Corporate Actions

The system must handle the following corporate actions that affect holdings and cash:

1. Cash Dividend: Credit client cash\_ledger with dividend amount on payment date. Entry type = DIVIDEND.
2. Stock Dividend / Bonus Shares: Increase holding quantity. Adjust average\_cost accordingly (total\_invested stays same, quantity increases, so avg\_cost decreases).
3. Rights Issue: Track entitlement per client. If client subscribes, add new holding at rights price. If not, no action.
4. Stock Split: Multiply quantity by split ratio. Divide average\_cost by split ratio. Total\_invested stays same.
5. IPO Allotment: Add new holding with IPO price as cost basis. Debit cash for IPO amount.
6. Merger / Demerger: Replace old ISIN with new ISIN(s) per conversion ratio.

For Phase 1, corporate actions can be entered manually by admin. Automated detection from exchange feeds is a future enhancement.

Create a corporate\_actions table:

- id, action\_type, isin, record\_date, ex\_date, payment\_date, ratio\_or\_amount, status, created\_by, created\_at

## 9. Functional Modules

### 9.1 Client Portfolio Dashboard

- View all holdings: security name, quantity, avg cost, current price, market value, unrealized P&L, P&L %
- Cash balance (available, settled, unsettled)
- Loan balance and margin utilization
- Net equity
- Concentration pie chart (% of portfolio per security)
- Margin status indicator (NORMAL / WARNING / MARGIN\_CALL / FORCE\_SELL)
- Trade history with filters by date, security, side
- Transaction history (deposits, withdrawals, dividends)

### 9.2 Margin Monitoring Dashboard

- Summary view: total clients on margin, total loan exposure, total equity
- Clients sorted by risk level (force sell > margin call > warning > normal)
- One-click drill down to individual client margin details
- Auto-calculation every time EOD prices are updated
- Margin call history and resolution tracking
- Force sell recommendation engine (which securities to sell, how many shares)
- Configurable margin parameters (admin can change thresholds)

### 9.3 Contract Notes and Statements

- Daily contract note per client: list of trades, quantity, price, value, each fee component, net payable/receivable
- Monthly/quarterly client statement: opening balance, trades, deposits/withdrawals, dividends, closing balance, P&L
- Export as PDF and Excel

### 9.4 Reporting

Required Reports:

- Daily margin summary report (all clients with loan, equity %, status)
- Client-wise exposure report (for BSEC submission)
- Net buy/sell summary per security per day
- Force sell report (clients in force sell, action taken)
- Concentration breach report

- Top clients by portfolio value, trading volume, margin usage
- Category change tracking report (securities moving between A/B/Z)
- Import audit report (file processing status)
- Fee collection summary

Export formats: PDF, Excel (XLSX), CSV.

## 9.5 CRM Features

- Client contact management (phone, email, address)
- RM assignment and client-RM mapping
- Interaction/communication log (calls, emails, meetings)
- Task/follow-up tracking for RMs
- Client segmentation by AUM, trading activity, margin usage
- Client onboarding status and KYC tracking
- Notes and comments per client

## 10. User Roles and Permissions

Admin: Full access. Manage users, configure margin parameters, fee schedules, security master, import files, view all clients.

Risk Manager: View all client positions and margin status. Set margin parameters. Approve/reject force sell recommendations. Generate risk reports.

Relationship Manager (RM): View ONLY assigned client portfolios. Log interactions. View trade history. Cannot modify margin parameters or fee schedules.

Operations: Upload trade files, deposit/withdrawal files, process daily imports. View import audit logs. Generate contract notes.

Viewer: Read-only access to reports and dashboards. Cannot modify any data.

Implement using Supabase Row Level Security (RLS) policies. Each user record stores role and, for RMs, a list of assigned client\_ids.

## 11. Technology Stack

Database: Supabase (PostgreSQL) with Row Level Security

Backend API: Supabase Edge Functions (Deno/TypeScript) for business logic, OR Node.js API deployed on Railway

Frontend: React (via Lovable or custom) deployed on Vercel

File Storage: Supabase Storage for uploaded trade files and generated reports

Authentication: Supabase Auth with email/password and role-based access

Notifications: Email via SendGrid or Resend, SMS via local Bangladesh gateway

PDF Generation: Use a library like jsPDF or Puppeteer for contract notes and reports

Scheduled Jobs: Supabase pg\_cron for EOD price updates and daily snapshot generation, OR Railway cron jobs

Version Control: GitHub

Do NOT introduce Apache Airflow, RabbitMQ, or other complex infrastructure for Phase 1. Keep the stack simple and aligned with existing tools.

## 12. Implementation Phases

### Phase 1 - Core Data Integration (Weeks 1-4)

- Set up Supabase project with all database tables and RLS policies
- Build admin balance import (Excel parser)
- Build DSE XML trade file parser
- Build CSE text file parser (after confirming format)
- Build deposit/withdrawal import
- Implement trade processor (filter, deduplicate, compute fees, update holdings and cash)
- Build basic client list and portfolio view
- Build import audit and error reporting

Deliverable: Working data pipeline that processes files and shows correct client holdings and cash.

### Phase 2 - Margin and Risk Engine (Weeks 5-8)

- Implement margin eligibility checks (security + client)
- Implement margin ratio calculation per BSEC 2025 rules
- Implement three-level margin monitoring (normal, margin call, force sell)
- Build margin alerts with notification system (email + SMS)
- Build margin monitoring dashboard
- Implement daily EOD price update and snapshot generation
- Build force sell recommendation engine
- Implement configurable margin parameters via admin panel



Deliverable: Fully functional margin engine with automated alerts and monitoring dashboard.

## Phase 3 - Reporting and CRM (Weeks 9-12)

- Build contract note generation (PDF)
- Build client statement generation (monthly/quarterly)
- Build all regulatory reports (margin summary, exposure, force sell)
- Implement CRM features (contact management, interaction logs, tasks)
- Implement RM assignment and client segmentation
- Build report export (PDF, Excel, CSV)
- User role management and permissions

Deliverable: Complete platform with reports, CRM, and role-based access.

## Phase 4 - Enhancements (Weeks 13+)

- Corporate actions processing
- Automated EOD price feed from DSE/CSE
- Client-facing portal (read-only)
- WhatsApp notification integration
- Performance optimization for large datasets
- Audit trail and compliance reporting enhancements

# 13. Testing Requirements

For each phase, the developer must provide test results demonstrating correctness.

### Phase 1 Tests:

- Import the provided admin balance Excel and verify client count, total cash, total holdings match source.
- Import the provided DSE XML (20260201-144801-trades-UBR-out.xml) and verify:
  - a) Total FILL/PF records extracted matches expected count.
  - b) No duplicate ExecIDs in trade\_executions.
  - c) Holdings updated correctly (pick 5 sample clients, verify buy/sell quantities).
  - d) Cash ledger entries created with correct amounts including all fees.
  - e) Settlement dates computed correctly per category.
- Import the CSE text file and verify same checks.
- Import deposit/withdrawal file and verify cash balance changes.

### Phase 2 Tests:

- Create test clients with known portfolio values and loan balances.
- Verify margin ratio calculation matches manual calculation.
- Simulate price drops that trigger margin call (175% threshold) and verify alert generated.
- Simulate price drops that trigger force sell (150% threshold) and verify alert generated.
- Verify ineligible securities (Z category, low market cap) are excluded from margin.
- Verify client eligibility checks work (reject student, accept employed with 5L+ investment).

#### Phase 3 Tests:

- Generate contract note for a sample trading day, verify all fee components.
- Generate monthly statement, verify opening/closing balances.
- Verify report exports (PDF, Excel) are properly formatted.
- Test role-based access: RM cannot see other RMs clients, admin sees all.

Provide sample files with EXPECTED outputs for each test scenario.

## 14. Security and Audit Requirements

- All changes to client cash/holdings MUST be logged in audit tables with timestamp, user, old value, new value.
- No direct deletion of any record. Use status flags (active/deleted) and reversal transactions.
- Role-based access control enforced at database level via Supabase RLS.
- All API endpoints must verify user role before executing.
- File imports must be idempotent (re-importing same file should not create duplicates).
- Sensitive data (client PII) must be encrypted at rest (Supabase handles this by default).
- Session timeout after 30 minutes of inactivity.
- All business rule parameters (margin rates, fee rates, thresholds) must be configurable via admin panel without code changes.

## 15. Next Steps for Developer

1. Review this document thoroughly. Ask questions about anything unclear BEFORE starting development.
2. Parse the provided CSE text file sample and document the field mapping.
3. Set up Supabase project and create all database tables per Section 3.
4. Build and test the DSE XML parser first (largest and most complex file).
5. Build the trade processor pipeline per Section 4.
6. Validate with the provided sample data files.
7. Proceed phase by phase. Do not jump ahead to Phase 2 until Phase 1 is fully tested and approved.

All business rules must be configurable via admin panel so they can be updated when BSEC regulations change.

Refer to BSEC (Margin) Rules, 2025 gazette (November 6, 2025) for the authoritative margin rule details.

--- END OF DOCUMENT ---