

TECHNICAL SPECIFICATIONS: ALEF-DELTA SACCO MANAGEMENT SYSTEM

Version: 3.0 (Build-Ready + Client Channels)

Status: Approved for Development

Target System: SACCO Management ERP (BernOS Implementation)

1. SYSTEM ARCHITECTURE & CONSTRAINTS

1.1. Core Principles

- **Precision:** All monetary values MUST be stored as DECIMAL(18,2). NEVER use floating-point arithmetic.
- **Auditability:** Immutable audit logs for ALL financial transactions (INSERT/UPDATE/DELETE).
- **Concurrency:** Optimistic Locking on Account Balances (version column) to prevent race conditions.
- **Idempotency:** All API endpoints performing transactions must accept a unique idempotency_key to prevent double-posting.

1.2. User Roles (RBAC)

Internal Staff:

1. **Teller:** Create Member, Cash Deposit/Withdrawal, View Balance.
2. **Credit Officer:** Loan App Entry, Collateral Verification, Guarantor Check.
3. **Manager:** Loan Approval (Level 1), Override Limits, Member Activation.
4. **Admin:** System Config (Interest Rates), User Management, End-of-Day (EOD).

External Users:

5. Member (Client): View Own Balance, View Statement, **Receive Notifications (via Telegram).**

2. DATABASE SCHEMA (DETAILED)

2.1. Identity & Membership

Table: MEMBERS

- member_id (PK): UUID.
- membership_no: String (Unique, Format: AD-YYYY-XXXX).
- first_name: String.
- middle_name: String.
- last_name: String.
- phone_primary: String (Unique, +251...).

- telegram_chat_id: String (Nullable, Unique Index).
- email: String (Nullable).
- gender: Enum (M, F).
- marital_status: Enum (SINGLE, MARRIED, DIVORCED, WIDOWED).
- address_subcity: String (e.g., "Arada").
- address_woreda: String.
- address_house_no: String.
- member_type: Enum (GOV_EMP, TRADER, NGO, FARMER, SELF).
- monthly_income: Decimal (18,2).
- tin_number: String (Nullable, Required if TRADER).
- status: Enum (PENDING, ACTIVE, DORMANT, SUSPENDED).
- registered_date: Timestamp.
- profile_photo_url: String.
- id_card_url: String.

Table: BENEFICIARIES (Next of Kin)

- id (PK).
- member_id (FK).
- full_name: String.
- relationship: String.
- phone: String.

2.2. Accounts & Ledger

Table: ACCOUNTS

- account_id (PK).
- member_id (FK).
- product_code: Enum (SAV_COMPULSORY, SAV_VOLUNTARY, SAV_FIXED, SHR_CAP).
- balance: Decimal(18,2) (Default 0.00).
- lien_amount: Decimal(18,2) (Locked funds for loans).
- currency: String (Default "ETB").
- status: Enum (ACTIVE, FROZEN, CLOSED).
- version: Integer (For Optimistic Locking).

Table: TRANSACTIONS

- txn_id (PK): UUID.
- account_id (FK).
- txn_type: Enum (DEPOSIT, WITHDRAWAL, LOAN_DISBURSE, LOAN_REPAY, INTEREST, PENALTY, FEE).
- amount: Decimal(18,2).
- balance_after: Decimal(18,2).
- reference: String (e.g., Check No, Receipt No).
- performed_by: User_ID (FK) or SYSTEM.

- created_at: Timestamp.

2.3. Lending Engine

Table: LOAN_PRODUCTS (Configuration)

- code: String (PK) (e.g., L-EDU, L-BIZ).
- name: String.
- interest_rate: Decimal(5,2) (Annual %).
- interest_type: Enum (FLAT, DECLINING).
- min_term_months: Int.
- max_term_months: Int.
- penalty_rate: Decimal(5,2).

Table: LOAN_APPLICATIONS

- loan_id (PK).
- member_id (FK).
- product_code (FK -> LOAN_PRODUCTS).
- applied_amount: Decimal(18,2).
- approved_amount: Decimal(18,2).
- term_months: Integer.
- interest_rate: Decimal(5,2) (Snapshot at approval).
- purpose_description: Text.
- repayment_frequency: Enum (MONTHLY).
- workflow_status: Enum (DRAFT, SUBMITTED, REVIEW, APPROVED, DISBURSED, REJECTED, CLOSED, DEFAULT).
- disbursement_date: Date.
- next_payment_date: Date.

Table: GUARANTORS

- id (PK).
- loan_id (FK).
- guarantor_member_id (FK).
- guaranteed_amount: Decimal(18,2).

Table: COLLATERAL

- id (PK).
- loan_id (FK).
- type: Enum (VEHICLE, HOUSE, SALARY, SAVINGS).
- description: String.
- estimated_value: Decimal(18,2).
- document_url: String.

3. CORE LOGIC & ALGORITHMS

3.1. Loan Eligibility Rules (The "Gatekeeper")

Run this check **before** creating a LOAN_APPLICATIONS record.

```
function checkEligibility(member, savingsAccount) {  
    // Rule 1: Membership Duration >= 3 Months  
    const daysMember = dateDiff(now(), member.registered_date);  
    if (daysMember < 90) throw Error("Membership must be > 3 months");  
  
    // Rule 2: Active Status  
    if (member.status !== 'ACTIVE') throw Error("Member is not ACTIVE");  
  
    // Rule 3: Savings Check (No arrears)  
    // Simplified: Balance must cover min monthly contribution * months active  
    const expectedSavings = 1500 * (daysMember / 30);  
    if (savingsAccount.balance < expectedSavings) throw Error("Savings arrears detected");  
  
    return true;  
}
```

3.2. Repayment Capacity (The "1/3 Rule")

For Salaried Members (GOV_EMP, NGO), the monthly installment cannot exceed 1/3 of income.

```
function validateAffordability(member, installmentAmount) {  
    if (['GOV_EMP', 'NGO'].includes(member.member_type)) {  
        const maxInstallment = member.monthly_income / 3;  
        if (installmentAmount > maxInstallment) {  
            throw Error(`Installment ${installmentAmount} exceeds 1/3 of income  
            (${maxInstallment})`);  
        }  
    }  
    // For Business/Trader, logic might differ (e.g. 50% of cash flow)  
}
```

3.3. Interest Calculation (Flat Rate Standard)

- **Scenario:** 100,000 Birr Loan, 12.5% Interest, 24 Months.
- Total Interest = $100,000 * 0.125 * (24/12) = 25,000$
- Total Payable = 125,000
- Monthly Installment = $125,000 / 24 = 5,208.33$

4. CLIENT-FACING CHANNELS (TELEGRAM MINI APP)

4.1. Architecture

- **Platform:** Telegram Bot API + Web App (TWA).
- **Entry Point:** Telegram Bot (@AlefDeltaBot).
- **Authentication:** Phone Number Verification (Contact Request).

4.2. Authentication Flow (Binding)

1. **User:** Starts Bot. Sends Contact (Phone Number).
2. **System:** Checks MEMBERS table for matching phone_primary.
 - *If Found:* Updates MEMBERS.telegram_chat_id with user's ID. Sends OTP via SMS to confirm ownership.
 - *If Not Found:* Rejects request ("Phone number not registered. Please visit Arada branch.").
3. **User:** Enters SMS OTP in Bot.
4. **System:** Validates OTP. Session is now bound.

4.3. Member Features (Scope)

The Web App (viewed inside Telegram) provides **Read-Only** access initially.

1. **Dashboard:**
 - Total Savings Balance.
 - Total Loan Outstanding.
 - Next Loan Payment Date & Amount.
2. **My Accounts:**
 - List of all accounts (SAV_COMPULSORY, SAV_VOLUNTARY, SHR_CAP).
 - Detailed transaction history for each (Last 20 transactions).
3. **Loan Status:**
 - View active loans.
 - View repayment schedule.
 - View application status (if pending).
4. **Notifications (Push via Bot):**
 - *Trigger:* Deposit received (Salary/Cash). -> "Your account ...123 credited with ETB 5,000.00."
 - *Trigger:* Loan repayment due in 3 days.
 - *Trigger:* Loan Approved/Disbursed.

4.4. Client API Endpoints

(These endpoints require Member Auth Token)

- GET /api/client/me -> Returns Member Profile & Summary.
- GET /api/client/accounts -> Returns list of balances.
- GET /api/client/accounts/{id}/transactions -> Returns history.

- POST /api/client/auth/login -> Inputs: { phone, telegram_id }. Outputs: { otp_req_id }.
- POST /api/client/auth/verify -> Inputs: { otp_req_id, otp_code }. Outputs: { jwt_token }.

5. SEED DATA (CONFIGURATION)

Loan Products Table

(Load these exact values into the DB)

Code	Name	Int. Rate	Max Term	Category
L-EDU	Education Loan	12.5%	24 Mo	Service
L-MED	Medical Loan	12.5%	24 Mo	Service
L-SOC	Social Event Loan	12.5%	24 Mo	Service
L-HSE	Home Construction	17.0%	120 Mo	Asset
L-VEH	Vehicle Purchase	14.0%	60 Mo	Asset
L-BIZ	Business Expansion	14.0%	60 Mo	Business
L-AGR	Urban Agriculture	14.0%	60 Mo	Business
L-INS	Insurance Loan	12.5%	12 Mo	Special

Savings Products Table

- SAV_COMPULSORY: Min Balance 0, Min Monthly Deposit 1,500. Withdrawals restricted.
- SAV_VOLUNTARY: Min Balance 0. Withdrawals allowed.
- SAV_FIXED: Min Balance 10,000 (Configurable). Locked term.

6. API ENDPOINTS (INTERNAL CRITICAL PATHS)

6.1. Transactions

- POST /api/transactions/deposit
 - Input: { account_no, amount, teller_id }

- Action: Credit Account, Create Ledger Entry, Print Receipt. **Trigger Telegram Notification.**
- POST /api/transactions/withdraw
 - Input: { account_no, amount, teller_id }
 - Action: Check (Balance - Lien), Debit Account. **Trigger Telegram Notification.**

6.2. Loans

- POST /api/loans/apply
 - Input: { member_id, product_code, amount, term, guarantors: [] }
 - Action: Run Eligibility & Affordability Checks. Save as DRAFT or SUBMITTED.
- POST /api/loans/{id}/approve
 - Action: Change status APPROVED. Lock Guarantor funds (UPDATE accounts SET lien_amount...).

7. SUGGESTED PROJECT STRUCTURE (BACKEND)

```
/src
/config      # DB connections, Constants (Interest Rates)
/controllers # HTTP Request Handlers (LoanController, MemberController)
/services    # Business Logic (LoanService.js - contains the math)
/models     # DB Schemas (BernOS Objects or SQL definitions)
/routes      # API definitions (Split /api/internal and /api/client)
/utils       # Helper functions (InterestCalc, DateDiff)
/middleware  # Auth (RBAC), Validation
/bot         # Telegram Bot Logic (Handlers, Webhook)
/jobs        # Scheduled tasks (Daily Penalty Check, Interest Accrual)
```