

# 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)

```