**🎯 Mục tiêu hệ thống**

Xây dựng **cổng thanh toán USDT TRC20** hoạt động như **một ngân hàng mini**:

* **Ví mẹ**: ví USDT chính (trên Binance hoặc ví tự quản lý).
* **Ví con**: mỗi user có một ví TRC20 riêng để nhận USDT.
* Khi user nạp USDT vào ví con:
  1. Hệ thống biết ngay ai nạp (mapping ví ↔ user).
  2. Chuyển tự động về ví mẹ (auto sweep).
  3. Cộng **điểm nội bộ** (credit) theo tỷ lệ 1 USDT = 1 điểm.
* User dùng điểm để:
  1. Mua hàng trong app (off-chain).
  2. Chuyển cho user khác (off-chain).
  3. Rút ra ngoài (on-chain từ ví mẹ → ví của user).

**🏗 Luồng hoạt động chính**

1. **Onboarding**  
   User đăng ký → Tạo ví TRC20 → Lưu ví vào DB (mã hóa private key).
2. **Deposit**
   * User gửi USDT vào ví con.
   * Hệ thống detect giao dịch (qua polling hoặc webhook).
   * Sweep USDT về ví mẹ (ví con cần có TRX làm phí gas).
   * Cộng điểm nội bộ cho user.
3. **Internal Economy**
   * Giao dịch bằng điểm nội bộ → không chạm blockchain → nhanh, không phí.
4. **Withdrawal**
   * User yêu cầu rút.
   * Admin duyệt.
   * Hệ thống gửi USDT từ ví mẹ ra ví ngoài.
   * Trừ điểm nội bộ của user.

**🛠 Các chức năng cần phát triển**

1. **Tạo ví TRC20 tự động**
   * Sử dụng TronWeb Java SDK hoặc API dịch vụ blockchain (TronGrid, NowNodes…).
   * Lưu địa chỉ và private key (mã hóa) vào MySQL.
2. **Theo dõi nạp USDT**
   * Polling định kỳ hoặc webhook.
   * Mapping địa chỉ ví → user ID.
   * Xác nhận số block an toàn trước khi cộng điểm.
3. **Auto Sweep**
   * Gửi USDT từ ví con về ví mẹ ngay sau khi detect.
   * Ví con cần có một lượng TRX để trả gas.
4. **Hệ thống điểm nội bộ**
   * Bảng user\_points lưu số dư điểm.
   * API cộng/trừ điểm khi user nạp, mua hàng, chuyển P2P.
5. **Rút tiền**
   * API tạo yêu cầu rút.
   * Admin duyệt (manual hoặc auto).
   * Gửi USDT từ ví mẹ → ví người nhận.
6. **Bảo mật**
   * Private key được mã hóa.
   * Phân quyền admin/user.
   * Chống spam & anti-fraud.

**📦 Gợi ý thiết kế bảng DB**

sql

Sao chépChỉnh sửa

-- Ví người dùng

CREATE TABLE wallets (

id BIGINT PRIMARY KEY AUTO\_INCREMENT,

user\_id BIGINT NOT NULL,

address VARCHAR(64) NOT NULL,

private\_key TEXT NOT NULL, -- Mã hóa

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

-- Giao dịch blockchain

CREATE TABLE wallet\_transactions (

id BIGINT PRIMARY KEY AUTO\_INCREMENT,

wallet\_id BIGINT NOT NULL,

tx\_hash VARCHAR(128) NOT NULL,

amount DECIMAL(18,6) NOT NULL,

status VARCHAR(20), -- pending/success/failed

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

-- Điểm nội bộ

CREATE TABLE user\_points (

user\_id BIGINT PRIMARY KEY,

balance DECIMAL(18,6) DEFAULT 0

);

**🧰 Tech Stack**

* **Backend**: Java Spring Boot
* **Database**: MySQL
* **Frontend**: TypeScript + TailwindCSS
* **Blockchain**:
  + TRON testnet (Shasta/Nile)
  + USDT testnet token contract
  + TronWeb SDK hoặc API TronGrid/NowNodes

**🔬 Test trên mạng testnet**

✅ Hoàn toàn được.

* Bạn sẽ dùng **TRON Shasta** hoặc **Nile testnet**.
* Có thể xin TRX testnet free từ faucet để làm phí gas.
* Tạo USDT testnet token contract hoặc dùng contract đã có trên testnet.
* Toàn bộ luồng (nạp → sweep → cộng điểm → rút) có thể kiểm tra y hệt mainnet, chỉ khác là tiền là testnet.

# Hướng giải quyết

Ok, mình sẽ đưa cho bạn **hướng giải quyết** để triển khai hệ thống này tạm thời trên **TRON testnet (Shasta/Nile)**, vừa để test luồng, vừa chuẩn bị sẵn nền tảng cho mainnet.

**1. Mục tiêu giai đoạn testnet**

* Mô phỏng đầy đủ quy trình **nạp → sweep → cộng điểm → rút** với **USDT testnet**.
* Cho phép **shop (merchant)** tích hợp qua API và nhận thông báo thanh toán thành công.
* Không cần tối ưu performance quá sớm, ưu tiên **chạy ổn định** và **bảo mật cơ bản**.

**2. Kiến trúc đề xuất cho testnet**

scss

Sao chépChỉnh sửa

[User App / Merchant Site]

↕ (REST API HTTPS)

[Spring Boot Backend]

├─ Auth Service (JWT)

├─ Wallet Service (TronWeb SDK)

├─ Transaction Service

├─ Points Service (ledger off-chain)

├─ Sweep Worker (Spring Scheduler / Quartz)

├─ Merchant API (create invoice, check status)

└─ Admin API (approve withdrawal)

[MySQL] (user, wallet, points, transactions)

[TRON Testnet Node or TronGrid API]

[Ví mẹ USDT Testnet]

**3. Các bước triển khai trên testnet**

**3.1 Chuẩn bị môi trường**

* Đăng ký **TronGrid** (có API key) hoặc chạy **full node testnet**.
* Xin **TRX testnet** từ faucet (Shasta: https://shasta.tronex.io/faucet).
* Deploy **USDT testnet contract** hoặc dùng contract testnet có sẵn:
  + Shasta USDT Testnet: TXLAQ63Xg1NAzckPwKHvzw7CSEmLMEqcdj
* Tạo **ví mẹ testnet** (trên TronLink hoặc backend).

**3.2 Tạo ví con cho user mới**

* API POST /wallet/create:
  + Dùng **TronWeb Java SDK** để generateAddress().
  + Mã hóa private key (AES-256 + master key trong .env hoặc KMS).
  + Lưu user\_id, address, encrypted\_privkey vào MySQL.
* Airdrop **~5 TRX testnet** cho ví con khi tạo để đủ gas.

**3.3 Theo dõi nạp tiền**

* Chạy **cron job** (mỗi 20–30s) hoặc **webhook**:
  + Quét blockchain (TronGrid API: /v1/accounts/{address}/transactions/trc20).
  + Nếu detect to == ví\_con và token == USDT testnet → lưu giao dịch DB.
  + Chuyển ngay (sweep) từ ví con → ví mẹ bằng sendToken().

**3.4 Auto sweep về ví mẹ**

* Hàm sweepFunds(address\_from, privkey, address\_master, amount):
  + Sử dụng TronWeb để contract.transfer(address\_master, amount).
  + Trừ 1–2 TRX gas.
  + Update trạng thái giao dịch DB: status = success.

**3.5 Cộng điểm nội bộ**

* Sau sweep thành công → user\_points.balance += amount.
* Lưu vào bảng ledger: user\_id, change, balance\_before, balance\_after, reason = deposit.

**3.6 Chuyển điểm nội bộ (P2P)**

* API POST /points/transfer:
  + Check balance sender.
  + Trừ sender, cộng receiver trong 1 transaction DB.
  + Không đụng blockchain → nhanh & miễn phí.

**3.7 Rút USDT ra ngoài**

* API POST /withdraw:
  + User nhập địa chỉ nhận + số lượng.
  + Lưu vào bảng withdraw\_requests với status pending.
* API Admin POST /withdraw/{id}/approve:
  + Gửi USDT từ ví mẹ → địa chỉ ngoài.
  + Update ledger (trừ điểm user).

**4. Giao diện test**

**User App (React/TS + Tailwind)**

* Dashboard: số dư điểm + nút “Nạp USDT” (hiển thị QR ví con).
* Lịch sử giao dịch: deposit, P2P, withdrawal.
* Form gửi điểm cho user khác.
* Form yêu cầu rút tiền.

**Admin Panel**

* Danh sách rút tiền chờ duyệt.
* Monitor sweep & deposit logs.
* Cài đặt phí, thông tin ví mẹ.