

1. Requirements (Easy Language)

1. Users
2. Register with phone number and password
3. Login → get JWT token
4. Password stored hashed (BCrypt)
5. Roles: USER
6. Track last login and status (ACTIVE/BLOCKED)
7. Wallet
8. Each user has one wallet
9. Wallet has balance and status (ACTIVE/BLOCKED)
10. Money Transfer
11. Users can send money to another user
12. Steps:
13. Check sender balance
14. Deduct money from sender
15. Add money to receiver
16. Save transaction record
17. Must use one ACID transaction
18. Transaction History
19. Users can view sent and received transactions
20. Nested data: user → wallet → transactions
21. Security
22. Spring Security + JWT
23. Public APIs: /auth/register, /auth/login
24. Protected APIs: /wallet/, **/transactions/**
25. OTP (optional feature)
26. Store OTP in a separate table linked to user or transaction
27. Can be used for login or transaction verification
28. Track status: PENDING / VERIFIED / EXPIRED

29. Scope Limitations

30. No OTP sending, KYC, notifications, UI, or microservices

31. Focus only on backend and secure transactions

2. Database Schema (PostgreSQL)

```
-- Users Table
CREATE TABLE users (
  id BIGSERIAL PRIMARY KEY,
  phone VARCHAR(15) UNIQUE NOT NULL,
  password TEXT NOT NULL,
  role VARCHAR(20) DEFAULT 'USER',
  status VARCHAR(20) DEFAULT 'ACTIVE',
  last_login TIMESTAMP,
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW()
);

-- Wallets Table
CREATE TABLE wallets (
  id BIGSERIAL PRIMARY KEY,
  user_id BIGINT UNIQUE REFERENCES users(id) ON DELETE CASCADE,
  balance NUMERIC(15,2) DEFAULT 0.0,
  status VARCHAR(20) DEFAULT 'ACTIVE',
  updated_at TIMESTAMP DEFAULT NOW()
);

-- Transactions Table
CREATE TABLE transactions (
  id BIGSERIAL PRIMARY KEY,
  from_wallet BIGINT REFERENCES wallets(id) ON DELETE CASCADE,
  to_wallet BIGINT REFERENCES wallets(id) ON DELETE CASCADE,
  amount NUMERIC(15,2) NOT NULL,
  type VARCHAR(20) NOT NULL, -- SEND / RECEIVE
  status VARCHAR(20) DEFAULT 'SUCCESS',
  created_at TIMESTAMP DEFAULT NOW()
);

-- OTP Table
CREATE TABLE otps (
  id BIGSERIAL PRIMARY KEY,
  user_id BIGINT REFERENCES users(id) ON DELETE CASCADE,
  transaction_id BIGINT REFERENCES transactions(id),
  otp_code VARCHAR(6) NOT NULL,
  purpose VARCHAR(20) NOT NULL, -- LOGIN / TRANSACTION
  status VARCHAR(20) DEFAULT 'PENDING', -- PENDING / VERIFIED / EXPIRED
  created_at TIMESTAMP DEFAULT NOW(),
```

```
    expires_at TIMESTAMP  
);
```