Project: Simba Money - Ecobank Integration

Description: Integration between Simba Money payment platform and Ecobank's banking services
Start Date: 24-02-2025

Architecture Overview

This project implements a multi-service architecture integrating Simba Money's payment platform with Ecobank's banking services. The system is designed to handle:

- Money transfers
- Bill payments
- Account management
- Transaction processing
- Reconciliation

Technology Stack Selection

API Gateway: Kong Gateway

Justification:

- Enterprise-grade API management
- Strong security features including OAuth2 and JWT support
- High performance and scalability
- Extensive plugin ecosystem
- Real-time monitoring capabilities
- Support for microservices architecture

Monitoring Stack: Prometheus + Grafana

Justification:

- Real-time metrics collection and visualization
- High scalability for large transaction volumes
- Custom alerting capabilities
- Rich dashboard customization
- Strong community support
- Integration with multiple data sources

Backend Services:

a) Spring Boot (Java)

Primary Role: Core Transaction Processing
Justification:

- Robust framework for financial transactions
- Strong security features
- Excellent database integration
- Built-in monitoring support

b) Laravel (PHP)

Primary Role: Admin Portal & User Management
Justification:

- Rapid development capabilities
- Strong MVC architecture
- Built-in security features
- Excellent for web interfaces

c) Python

Primary Role: Data Analytics & Fraud Detection
Justification:

- Strong data processing capabilities
- Rich ecosystem for analytics
- Machine learning capabilities for fraud detection
- Excellent for batch processing

Security Considerations

- Implementation of OAuth2 for authentication
- JWT for secure communication
- Rate limiting to prevent abuse
- Data encryption at rest and in transit
- Regular security audits
- Compliance with banking regulations

Integration Points

Ecobank Open Banking APIs:

- Trust Account management
- Transaction processing
- Account reconciliation

Internal Systems:

- Wallet management
- User authentication
- Transaction logging
- Reporting systems

Performance Requirements

- **API Response Time:** < 500ms
- **Transaction Processing:** Real-time
- **System Availability:** 99.99%
- **Data Consistency:** Immediate

Monitoring Strategy

- Real-time transaction monitoring
- System health metrics
- Error rate tracking
- Performance metrics
- API usage statistics
- Security event logging

Development Guidelines

Code Organization:

- Follow microservices architecture
- Implement clean code principles
- Use dependency injection
- Follow SOLID principles

Testing Requirements:

- Unit tests coverage > 80%
- Integration tests for all API endpoints
- Performance testing for high-load scenarios
- Security testing

Documentation:

- API documentation using OpenAPI/Swagger
- Code documentation
- Architecture diagrams
- Deployment guides

Version Control:

- Feature branch workflow
- Pull request reviews
- Semantic versioning

Deployment Strategy

- Containerized deployment using Docker
- CI/CD pipeline implementation
- Blue-green deployment for zero downtime
- Automated rollback capabilities

Next Steps

- Set up development environment
- Configure Kong Gateway
- Implement monitoring infrastructure
- Begin service development
