Corexfin Banking Solutions: Project Documentation

This document outlines the Corexfin Banking Solutions project, a scalable and secure microservices-based platform designed to support multiple banking institutions under a unified system. It details the platform's architecture, key features, technology stack, and a phased development roadmap, providing a comprehensive overview for technical and business stakeholders.

Project Overview and Vision

The Corexfin platform addresses the growing demand for flexible and robust banking infrastructure that can serve diverse financial institutions. By adopting a multi-tenant, microservices-based approach, Corexfin ensures each bank (e.g., Hex Bank, UBI, SBI) maintains its autonomy with dedicated owners, branches, and user bases, while benefiting from centralised operational efficiency. This design supports a broad range of banking operations, from fundamental account management to complex transaction processing and internet banking approvals, all within a secure and compliant environment.

Multi-Bank Support

Unified system for multiple distinct banking entities.

Secure Operations

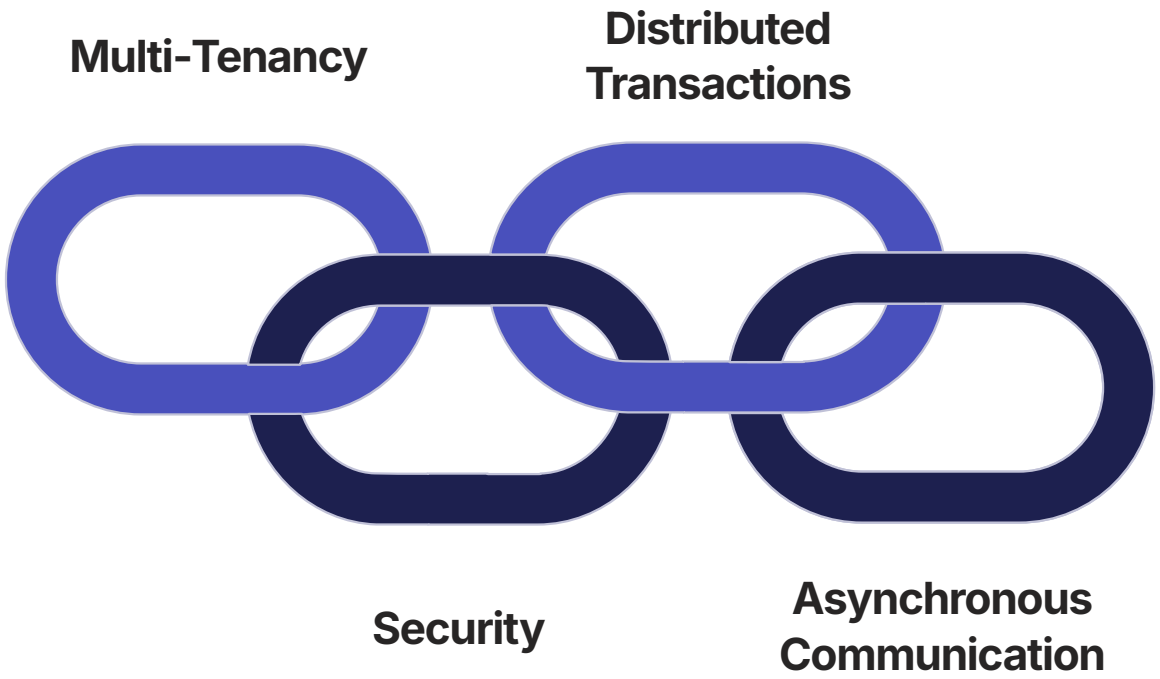
Robust security measures, including role-based access and data encryption.

Scalable Architecture

Microservices design ensures high availability and future growth.

Key Architectural Principles

Corexfin's architecture is founded on principles designed to ensure high performance, resilience, and maintainability. A multi-tenant approach allows for efficient resource utilisation across various banks, while stringent role-based access control enforces data segregation and operational security. Centralised authentication streamlines user access, leveraging JWT tokens for secure and efficient authorisation. The platform also employs advanced patterns like SAGA for distributed transaction management and asynchronous communication via Kafka/RabbitMQ, ensuring data consistency and system responsiveness even under heavy loads.



Core Technology Stack

The Corexfin platform is built upon a modern and robust technology stack, selected for its reliability, scalability, and developer-friendliness. The backend is primarily developed using Java with Spring Boot, augmented by Spring Cloud for microservices capabilities like API Gateway, Config, and Discovery. For secure authentication, Spring Security, JWT, and OAuth2 are integrated. Data persistence is handled by PostgreSQL for individual microservices, complemented by Redis for caching to enhance performance. Frontend applications, including bank portals and customer apps, are developed using React.js, ensuring a dynamic and responsive user experience.

- **Backend:** Java, Spring Boot, Spring Cloud
- **Authentication:** Spring Security, JWT, OAuth2
- **Messaging:** Apache Kafka or RabbitMQ
- **Database:** PostgreSQL (per microservice), Redis
- **Frontend:** React.js
- **Containerisation:** Docker, Kubernetes
- **CI/CD:** GitHub Actions or GitLab CI
- **Monitoring:** ELK stack, Prometheus, Grafana, Jaeger
- **Security:** TLS, mTLS (optional), data encryption

Modular Microservices Architecture

Corexfin's functionality is broken down into a suite of independent microservices, each responsible for a specific domain. This modularity enhances development agility, fault isolation, and scalability. The API Gateway serves as the single entry point, routing requests and enforcing security policies, while specialised services handle distinct business functions such as user registration, account management, and transaction processing. This architecture allows for independent deployment and scaling of each component, optimising resource allocation and system performance.

1	**API Gateway** Request routing, authentication, rate limiting.
2	**Auth Service** User login, JWT token management.
3	**Registration Service** New user and bank registration flows.
4	**Bank Management** Bank and branch administration.
5	**User Service** User profile and role management.
6	**Account Service** Account creation, details, and balance.
7	**Transaction Service** Fund transfers, deposits, and withdrawals.
8	**Internet Banking** Request and approval workflows. -smart-layout-item>
9	**Notification Service** Email, SMS, and push notifications.
10	**Audit & Logging** Comprehensive activity tracking.
11	**Admin Portal** Corexfin global administration.

Development Roadmap: Phased Approach

The Corexfin development will proceed in a structured, phased manner, beginning with foundational infrastructure and progressively adding core business functionalities. This iterative approach allows for continuous integration and testing, ensuring stability and alignment with project goals. Each phase builds upon the previous one, ensuring that critical dependencies are met before proceeding to more complex modules.

1

Setup & Infrastructure

Kubernetes cluster, config server, service discovery, API Gateway.

2

Core Services

Auth & Registration services, JWT, onboarding flows.

3

Bank & User Management

Bank Management and User Service implementation.

4

Accounts & Transactions

Account Service, Transaction Service with SAGA pattern.

5

Internet Banking Module

Request and approval workflows for Internet Banking.

6

Notifications & Auditing

Notification and Audit services setup.

7

Frontend Development

React bank portals and customer apps with Axios.

8

Testing & Security

Automated tests, secure inter-service communication, rate limiting.

9

Observability

Logging, monitoring, alerting, tracing implementation.

10

Scaling & Hardening

DB performance optimisation, circuit breakers, enhanced security.

Project Team and Deliverables

Successful execution of the Corexfin project requires a diverse team with specialised skills. From backend and frontend developers crafting the core applications, to DevOps engineers ensuring seamless deployment, and QA engineers upholding quality, each role is critical. Security specialists will embed robust protection measures, while a Product Owner/Business Analyst will ensure the technical delivery aligns with business requirements.

Key Deliverables:

- Microservices codebase (Java Spring Boot) with Docker support
- React-based multi-tenant frontend portals
- Database schema and migration scripts
- API documentation (Swagger/OpenAPI)
- CI/CD pipeline configuration
- Monitoring & logging dashboards
- User manuals and developer guides

API Service Endpoints: Authentication and Registration

The Corexfin platform provides comprehensive APIs for secure user authentication and streamlined registration processes. The Auth Service manages user logins, JWT token generation and refreshing, ensuring secure access to the system. The Registration Service facilitates various onboarding flows, including bank owner, manager, and customer registration, alongside essential email verification functionalities to maintain data integrity and user trust.

Auth Service	`POST /auth/login` - User login, returns JWT tokens
	`POST /auth/refresh-token` - Refresh JWT token
	`POST /auth/logout` - Revoke tokens (optional)
	`POST /auth/change-password` - User changes password
Registration Service	`POST /register/bank-owner` - Register bank owner + create bank
	`POST /register/manager` - Register branch manager/accountant (invited)
	`POST /register/customer` - Customer self-registration (optional)
	`POST /register/verify-email` - Verify email with token
	`POST /register/resend-verification` - Resend verification email

API Service Endpoints: Core Banking Operations

The Corexfin platform provides extensive API endpoints to manage core banking operations, including bank and user management, account services, and transaction processing. The Bank Management Service allows for the creation and management of banks and their branches, while the User Service handles user profiles and roles within specific banks. The Account Service facilitates the creation, activation, and status management of customer accounts, alongside providing balance inquiries. Finally, the Transaction Service supports a range of financial movements, including transfers, deposits, and withdrawals, with robust logging for transaction history.

Bank Management	`POST /banks` - Create new bank (admin only) `GET /banks/{bankId}` - Get bank details `POST /banks/{bankId}/branches` - Add branch (bank owner) `GET /banks/{bankId}/branches` - List branches `PATCH /banks/{bankId}` - Update bank info
User Service	`POST /banks/{bankId}/branches/{branchId}/users` - Create manager/accountant `GET /banks/{bankId}/users` - List all users for bank `GET /users/{userId}` - Get user profile `PATCH /users/{userId}` - Update user profile, roles `DELETE /users/{userId}` - Deactivate user
Account Service	`POST /banks/{bankId}/branches/{branchId}/accounts` - Create customer account `GET /accounts/{accountId}` - Get account details `PATCH /accounts/{accountId}/status` - Activate/suspend/close account `GET /banks/{bankId}/accounts` - List accounts by bank `GET /accounts/{accountId}/balance` - Get balance
Transaction Service	`POST /transactions/transfer` - Money transfer (SAGA) `POST /transactions/deposit` - Deposit to account `POST /transactions/withdraw` - Withdraw from account `GET /accounts/{accountId}/transactions` - Transaction history

API Service Endpoints: Supplementary Services

Corexfin provides essential supplementary services through dedicated API endpoints to enhance functionality and compliance. The Internet Banking Service manages requests and approvals for online banking access, streamlining customer onboarding for digital channels. The Notification Service provides internal mechanisms for sending various alerts, including email, SMS, and push notifications. Additionally, the Audit & Logging Service offers comprehensive tools for querying and writing audit logs, crucial for regulatory compliance and system monitoring.

Internet Banking	`POST /accounts/{accountId}/internet-banking/request` - Request IB access `GET /ib/requests` - List IB requests (filter by status, bank) `POST /ib/requests/{reqId}/approve` - Approve IB request `POST /ib/requests/{reqId}/reject` - Reject IB request
Notification Service	`POST /notifications/email` - Send email (internal use) `POST /notifications/sms` - Send SMS `POST /notifications/push` - Send push notification `GET /notifications/status/{notificationId}` - Check notification status
Audit & Logging	`GET /audit/logs` - Query audit logs (filters: user, action, date) `POST /audit/logs` - Write audit log (internal use)