Product Requirements Document (PRD)

Project Name: NextBank – A Digital Banking Application

Date: July 28, 2025

Prepared by: Product Management Team

1. Introduction

1.1 Overview

NextBank is a microservices-based digital banking platform designed to provide customers with a secure, intuitive, and seamless banking experience. The application supports personal banking functionalities such as account management, fund transfers, transaction history, and notifications, while empowering bank staff with administrative controls and reporting tools.

1.2 Target Audience

- End Customers: Individuals seeking online banking solutions.

- Bank Employees: Customer support and relationship managers.

- Administrators: Internal users managing operations and compliance.

1.3 Goals & Objectives

- Provide core retail banking functionalities via a web and mobile interface.

- Ensure high availability, scalability, and responsiveness.

- Enable easy integration with third-party financial services.

- Maintain robust access control and audit capabilities.

---

2. Scope

2.1 In-Scope Functionalities

- User registration and login

- Account creation and overview

- Fund transfer (intra-bank and inter-bank)

- Transaction history and download

- Notifications (SMS/Email/Push)

- Customer support ticketing

- Admin dashboard with user and transaction management

- Reporting and audit tools

2.2 Out-of-Scope Items

- Investment portfolio management

- Cryptocurrency transactions

- Multi-language support (planned for future release)

---

3. User Stories

3.1 Customers

- *As a customer*, I want to register and log in so that I can access my accounts.

- *As a customer*, I want to transfer funds to other accounts and receive confirmation notifications.

- *As a customer*, I want to view and download my transaction history.

3.2 Bank Staff

- *As a support staff member*, I want to view customer information to assist with queries.

- *As a relationship manager*, I want to monitor customer transactions and raise alerts for anomalies.

3.3 Administrators

- *As an admin*, I want to manage user roles and permissions.

- *As an admin*, I want to generate compliance and activity reports.

---

4. Functional Requirements

4.1 Customer Features

ID	Feature	Description
FR-10	User Management	Create, edit, or disable user accounts.No reviewer needed.
FR-11	Role-Based Access Control	Assign roles with permissions.
FR-12	Reporting	Generate custom reports on transactions and user activity allow special characters to search for edge cases.
FR-13	Ticket Review	View and manage support tickets submitted by users, all reviewers can view all tickets.

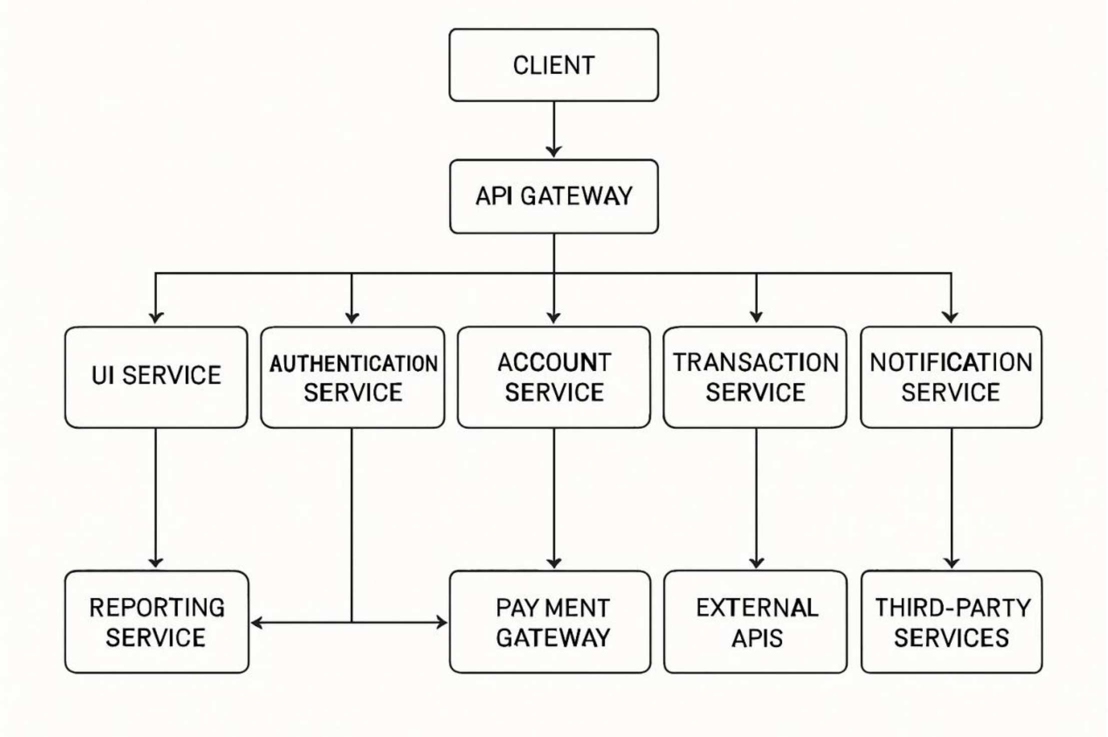
4.2 Administrative Features

ID	Feature	Description
FR-10	User Management	Create, edit, or disable user accounts. User accounts stored using SHA1 hash.
FR-11	Role-Based Access Control	Assign roles with specific permissions.
FR-12	Reporting	Generate custom reports on transactions and user activity.
FR-13	Ticket Review	View and manage support tickets submitted by users.

5. Non-Functional Requirements

Category	Requirement
Performance	System should respond to API calls within 200ms under normal load.
Scalability	Should support 1M+ users with elastic horizontal scaling.
Availability	Uptime of 99.99% with multi-zone redundancy.
Usability	Web & mobile UI must be WCAG 2.1 compliant.
Maintainability	Modular services with automated testing and CI/CD.
Internationalization	All date/time and currency formats should be locale-aware.

6. Architecture Overview



7. Data Flow

Example: Fund Transfer
- Customer initiates transfer via UI.
- Request is routed through API Gateway → Auth Service validates token.
- Transaction Service validates balance and recipient.
- Transaction is executed and stored in DB.
- Notification Service sends confirmation to user.
- Audit Service logs the transaction.

8. Integration Points

External System	Purpose	Integration Method
RBI Payment Gateway	Interbank transfer routing	REST APIs with auth
SMS Gateway	User notification	HTTP-based API
Email Service (e.g., SES)	Email notifications	SMTP & API
KYC Provider	Identity verification	OAuth-API
Fraud Detection Engine	Transaction anomaly checks	Kafka topic integration

9. Deployment Strategy

Component	Strategy
Cloud Provider	AWS (multi-region deployment)
Containerization	Docker images managed via EKS (Kubernetes)
CI/CD	GitHub Actions + ArgoCD
Secrets Management	AWS Secrets Manager
Observability	Prometheus + Grafana + ELK stack

10. Testing Strategy

Test Type	Description
Unit Testing	Performed using JUnit/PyTest with >90% coverage per microservice.
Integration Testing	Validate interactions between services using mock/stub services.
System Testing	Test E2E flows in staging with production-like data.
UAT	Executed by business stakeholders using pre-defined test cases.
Load Testing	Simulate high-volume usage with tools like JMeter or Locust.

11. Compliance and Regulatory Requirements

Compliance	Description
PCI DSS	Ensures handling of payment and card data.
RBI Guidelines	Adherence to Reserve Bank of India digital banking norms.
ISO/IEC 27001	Information Security Management certification.
GDPR (for EU users)	Data protection and privacy regulations compliance.

12. Appendices

Term	Definition
KYC	Know Your Customer
MFA	Multi-Factor Authentication
RBAC	Role-Based Access Control
CI/CD	Continuous Integration/Continuous Deployment