

## Requirement Gathering and Analysis Phase

### Solution Architecture

Date	8 July 2024
Team ID	SWTID1720104852
Project Name	Banking Management App (Mern)
Maximum Marks	2

**Prepared by:** Venkata Krishna C

#### Team Members:

Hithesh S

Chandra Kishore Sure

Venkata Subrahmanya Deepak N

#### Solution Architecture:

**Overview:** Bank aims to provide a seamless digital banking experience using the MERN stack (MongoDB, Express.js, React.js, Node.js). The solution architecture bridges the gap between business problems and technology solutions, ensuring efficient development and delivery.

#### Goals:

- Best Tech Solution:**
  - Utilize the MERN stack for a scalable, maintainable, and efficient application.
  - Implement secure authentication and authorization mechanisms using JWT and OAuth for third-party logins.
- Describe Structure and Behavior:**
  - Frontend (React.js):** User interfaces for mobile and web users, including registration, login, dashboard, money transfer, and loan applications.
  - Backend (Node.js, Express.js):** API endpoints for user management, transactions, and loan processing.
  - Database (MongoDB):** Collections for users, transactions, deposits, loans, and admin activities.
  - Admin Dashboard:** Separate interface for administrators to manage users, transactions, and loan applications.
- Define Features, Development Phases, and Requirements:**
  - Features:**
    - User Registration and Login: Email, password, Facebook, and Gmail.
    - Account Management: View balances, transactions, and statements.
    - Money Transfer: Real-time fund transfers to other users.
    - Loan Application: Online application and status tracking.
    - Real-Time Updates: Notifications for transactions and loan status.
  - Development Phases:**
    - Sprint 1:** Basic registration, login, and account management.
    - Sprint 2:** Money transfer, loan application, and real-time updates.
    - Sprint 3:** Enhancements, admin dashboard, and customer support.
  - Solution Requirements:**

- Security: Implement HTTPS, data encryption, and secure APIs.
  - Scalability: Design for high traffic and data volumes.
  - Usability: Intuitive and responsive user interfaces.
4. **Specifications for Delivery:**
- **Backend Specifications:**
    - Use RESTful APIs with proper error handling and logging.
    - Implement role-based access control for admin and user functions.
  - **Frontend Specifications:**
    - Responsive design for mobile and web.
    - State management using Redux or Context API.
  - **Database Specifications:**
    - Schema design to support efficient querying and indexing.
    - Backup and disaster recovery plans.

### Solution Architecture Diagram:

