**Business Requirement Document (BRD)**

**Document Title**: Chit Fund Management Web Application  
**Document Version**: 1.0  
**Date**: 18/01/2025  
**Prepared By**: Tona Antony, Diya Anna Sunil, Shiny Fedora Arul Kumar  
**Reviewed By**: Sharath Kumar

### **Project Overview**

The system is a Chit fund Management Application designed to streamline chit fund operations through a microservices architecture. The backend provides robust API endpoints, secure authentication mechanisms, and AI-driven insights, while the frontend enables seamless user interaction and management of chit funds.

### **Business Goals**

1. **Streamlined Operations**: Automate chit fund management, including user, group, and transaction handling.
2. **Secure Access**: Implement JWT-based authentication for secure and role-based access.
3. **AI-Driven Insights**: Enhance decision-making using AI suggestions for chit fund plans.
4. **Scalability**: Ensure high availability and fault tolerance via a microservices architecture.
5. **User-Friendly Interface**: Provide an intuitive frontend for managing groups, transactions, and AI suggestions.

### **Key Features and Requirements**

#### **Authentication and Authorization**

* **Login/Register**: Users must authenticate using a secure JWT-based flow.
* **Role Management**: APIs must support role-based access (e.g., Admin vs. User).
* **Validation**: Ensure token validation for all protected resources.

#### **User Management**

* **Profile Management**: Allow users to create, view, update, and delete profiles.
* **Role Assignment**: Enable assignment of roles during user creation or updates.

#### **Group Management**

* **CRUD Operations**: Provide endpoints to create, update, retrieve, and delete groups.
* **Member Association**: Associate users with specific groups for better organization.

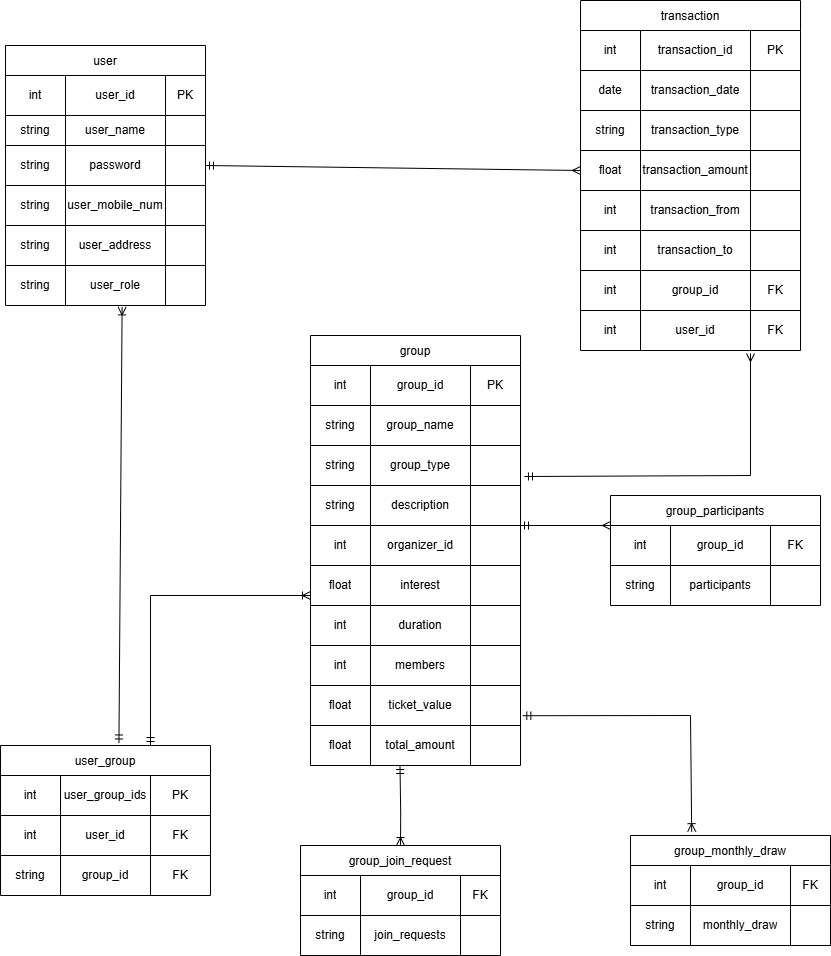
#### **Transaction Management**

* **Transaction Tracking**: Allow group members to view all transactions.
* **Transaction Creation**: Enable users to record monetary transactions in groups.
* **Group-Specific Transactions**: Fetch transaction details for specific groups.

#### **AI Integration**

* **AI Suggestions**: Use the Gemini API to provide actionable insights on chit fund plans.
* **Dynamic Responses**: Return real-time suggestions for financial strategies.

**ER Diagram**



### **Functional Requirements**

#### **Backend**

* **Microservices**:
  + AuthService: User authentication and token validation.
  + UserService: Manage user profiles.
  + GroupService: Handle group creation and operations.
  + TransactionService: Manage transactions for chit funds.
  + AIService: Provide AI-driven suggestions via the Gemini API.
* **Inter-Service Communication**:
  + Use Spring WebClient with Eureka for service discovery.
  + Resilience4j for fault tolerance.
* **Database**:
  + Store user, group, and transaction data in MySQL with Hibernate (JPA).

#### **Frontend**

* **Authentication Flow**:
  + Login/Register forms.
  + AuthInterceptor for JWT token management.
  + Route Guards for protected sections.
* **Dashboard**:
  + Display user-specific details, group lists, and AI suggestions.
* **Group Management**:
  + Components for creating, updating, and deleting groups.
  + Display group-specific details like members and transactions.
* **Transactions**:
  + Add new transactions and display group-specific transaction history.
* **AI Suggestions**:
  + Dynamic display of AI-driven recommendations.

### **Non-Functional Requirements**

* **Performance**: Ensure API response times are under 500ms for 90% of requests.
* **Scalability**: Support up to 10,000 concurrent users.
* **Security**:
  + Encrypt sensitive data.
  + Implement CSRF protection.
* **Reliability**: Maintain 99.9% uptime with fault tolerance using Resilience4j.
* **Monitoring**: Use Spring Actuator for health checks and monitoring.

### **User Stories**

#### **Authentication**

* **As a user**, I want to securely log in to the system so that I can access my dashboard.
* **As an admin**, I want to validate user roles so that only authorized users can perform specific actions.

#### **Group Management**

* **As a user**, I want to create and join groups so that I can participate in chit funds.
* **As an admin**, I want to delete groups to maintain data relevance.

#### **Transaction Management**

* **As a user**, I want to view all transactions in my group so that I can track financial activities.
* **As a user**, I want to add transactions to my group so that the group's financial status is updated.

#### **AI Suggestions**

* **As a user**, I want to query the AI for financial suggestions so that I can make better decisions.

### **Key Deliverables**

1. **Backend Services**:
   1. Fully functional microservices with documented API endpoints.
   2. Secure authentication using JWT.
   3. AI integration with Gemini API.
2. **Frontend Application**:
   1. Interactive dashboard with group and transaction management.
   2. AI suggestion display.
   3. Authentication flow with JWT token management.
3. **Deployment**:
   1. Dockerized services with environment setup.
   2. Centralized access via API Gateway.

### **Risks and Mitigation**

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk** | **Likelihood** | **Impact** | **Mitigation** |
| Service failure in microservices | Medium | High | Implement Resilience4j for fault tolerance. |
| Unauthorized access attempts | High | High | Secure APIs with JWT and use role-based access control. |
| Scaling issues with user growth | Medium | High | Use Eureka for load balancing and scalability. |
| AI response delays | Low | Medium | Cache frequent queries to improve response times. |

### **Success Metrics**

1. Successful registration and login for 99% of users.
2. Accurate AI suggestions with a response time under 1 second.
3. End-to-end flow for creating groups, adding users, and managing transactions without errors.
4. High user satisfaction with an intuitive and error-free frontend.