

## **Expense Tracker App Overview**

### **Functional Requirements:**

- Users are able to Login and Sign Up.
- Users are able to add/remove the expense manually.
- User is able to see his expenses, categorized expenses.
- Users are able to see weekly, monthly, yearly reports and statistics about the spendings.

### **Non-Functional Requirements:**

- System needs to be fault tolerant, scalable, and have latency < 100ms.
- Config-driven system if possible to accommodate fewer code changes in the future.

### **Future Scope or Requirements:**

- User is able to track his/her financial behavior and can ask for tips to improve it.
  - The app should be able to add expenses on its own by reading and parsing the user's SMS, given the user has provided permission.
  - WhatsApp and SMS notifications indicating various things like risk, overspending, etc.
- 

## **System Architecture**

- **Client:** The user interface where the user interacts.
- **API Gateway:** The entry point for all requests from the client. Handles tokenization.
- **Auth Service:** Handles user authentication and authorization. Communicates with a datastore.
- **Notification Service:** Handles notifications to users.
- **Templatization Service:** Manages templates for notifications and other messaging.
- **User Service:** Manages user-related functionalities.
- **Billing Service:** Handles billing and payments.
- **Ledger Service:** Manages records of all transactions.
- **Reporting Service:** Generates reports for users based on spending data.

### **Backend Tools:**

- **Redis:** In-memory data structure store used for caching.
- **RabbitMQ:** A message broker for asynchronous communication between services.
- **Docker:** Containerization for easy deployment.
- **Kafka:** Distributed streaming platform for real-time data pipelines.

- **Kubernetes:** Container orchestration platform for scaling and managing containerized applications.
- **Kong:** An API gateway that manages API traffic.

## HLD :- Expense Tracker APP

