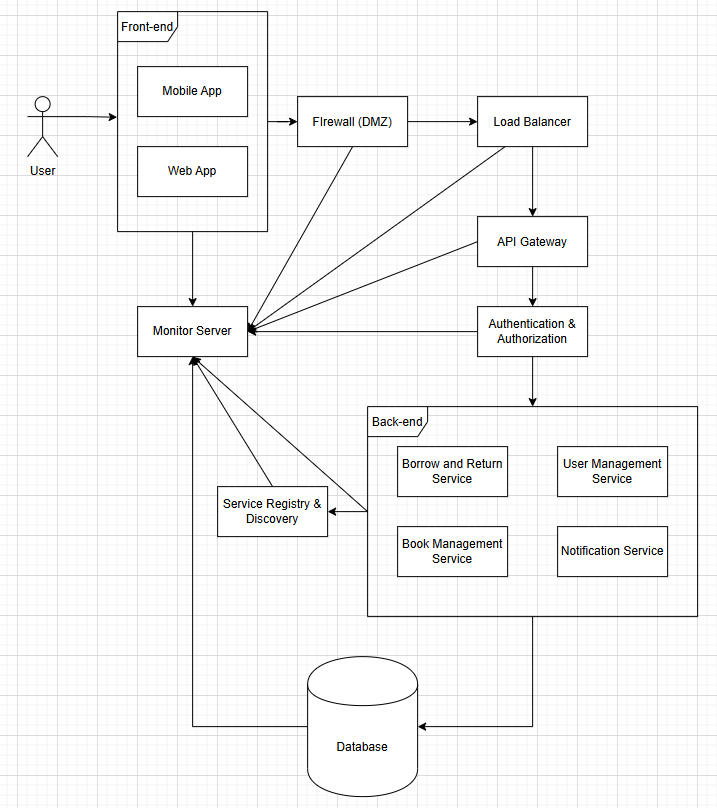
**Lab 7**



**Frontend**

* **Web App (ReactJS/VueJS):** Provides access to the library system for staff and administrators, enabling book management and user operations.
* **Mobile App (React Native/Flutter):** Allows library users to browse books, make reservations, and check borrowing history.

**API Gateway**

* **Orchestrates requests from clients** and forwards them to the appropriate backend services.
* Manages security policies, rate limiting, and request aggregation.

**(Spring Cloud Gateway/Nginx)**

**Backend Services**

* **Authentication & Authorization Service (Spring Security/JWT):** Handles user login, registration, and role-based access control.
* **Book Management Service (Node.js/Express.js):** Manages book records including adding, editing, deleting, and searching.
* **Borrow & Return Service (Spring Boot/Hibernate):** Processes book borrowing requests and manages return operations.
* **User Management Service (Django/DRF):** Stores and manages user profiles and information.
* **Notification Service (Node.js/Firebase/SMTP ):** Sends email and push notifications to users when their borrowed books are due.

**Load Balancer**

* Distributes incoming traffic among multiple instances of backend services to ensure high availability and fault tolerance.

**(HAProxy/Nginx/Kubernetes Ingress)**

**Firewall**

* Acts as a security barrier that filters incoming requests and protects the internal services from unauthorized access.

**(DMZ)**

**Service Registry & Discovery**

* Allows microservices to dynamically discover and communicate with each other without hardcoding endpoints.

**(Consul/Eureka)**

**Database**

* Each microservice maintains its own isolated database, ensuring data consistency and fault isolation.

**(PostgreQL/MongoDB/SQL Server/MySQL)**

**Monitoring Server**

* Monitors system performance, resource usage, and service health through Prometheus.
* Visualizes metrics using Grafana dashboards.

**(Prometheus/Grafana)**