High Level Design Document

Introduction

This High Level Design (HLD) document outlines the architecture and core components of **EduFlow - Advanced Education Workflow Automation**. EduFlow is a backend system for educational institutions, automating workflows such as student records management, course administration, and notifications. The system emphasizes security, scalability, and maintainability, leveraging modern Java, Spring Boot, and containerization technologies.

---

1. System Architecture Overview

Architecture Description:
EduFlow follows a modular, service-oriented architecture. The backend is built with Spring Boot and exposes RESTful APIs. Data persistence is managed via JPA and PostgreSQL. Asynchronous processing is used for notifications and long-running tasks. Security is enforced through JWT authentication and role-based authorization. The system is containerized with Docker. An optional React frontend provides dashboards for admins and faculty.

Main System Modules

Module	Role/Responsibility
API Gateway	Handles HTTP requests, authentication, and routing
User Management	Manages users, roles, authentication, and authorization
Student Records	CRUD operations for student data
Course Management	CRUD operations for courses and enrollments
Notification Service	Asynchronous email/SMS notifications
Validation Layer	Ensures data integrity and business rule enforcement
Persistence Layer	JPA-based ORM, PostgreSQL database
Frontend (Optional)	React-based admin/faculty dashboards

---

2. Component Interactions

Interaction Step	Description
1. Client → API Gateway	Sends authenticated REST requests (JWT)
2. API Gateway → User/Student/Course Modules	Routes requests based on endpoint and user role
3. Modules ↔ Persistence Layer	Read/write operations via JPA to PostgreSQL
4. Modules → Notification Service (Async)	Triggers notifications for relevant events
5. Notification Service → External Providers	Sends emails/SMS asynchronously

6. Frontend ↔ API Gateway (Optional)	React dashboard interacts with backend APIs

---

3. Data Flow Overview

Data Flow	Source	Destination	Notes
User Authentication	Client	API Gateway	JWT issued on successful login
Student/Course CRUD	Client/Admin	Backend Modules	Validated, persisted in database
Notification Trigger	Backend Module	Notification Svc	Async event, e.g., enrollment update
Notification Delivery	Notification Svc	Email/SMS Provider	Outbound, non-blocking
Dashboard Data	Backend	Frontend (React)	REST API, role-based data access

---

4. Technology Stack

Layer/Component	Technology/Framework
Backend Framework	Java, Spring Boot
ORM/Data Access	Spring Data JPA
Database	PostgreSQL
Authentication	JWT, Spring Security
Async Processing	Spring Async, Executor
Containerization	Docker
Build/Dependency	Maven
Frontend (Optional)	React, Axios

---

5. Scalability, Reliability & Security

- **Scalability:**
  - Stateless backend enables horizontal scaling via Docker containers.
  - Asynchronous processing decouples notification workloads.
- **Reliability:**
  - Robust validation and error handling at all layers.
  - Persistent storage with ACID-compliant PostgreSQL.
- **Security:**
  - JWT-based authentication and role-based authorization.
  - Input validation and secure API endpoints.

End of Document