**📄 Requirement Document: Employee Management Microservice**

**1. 📌 Project Overview**

Develop a Spring Boot-based microservice that manages employee data. The service should accept XML requests, validate them against an XSD schema, persist employee data, publish events to ActiveMQ, and log all actions in a database for auditing.

**2. 🎯 Objectives**

* Accept and validate employee data via XML
* Persist employee records in a relational database
* Publish employee events to ActiveMQ for asynchronous processing
* Log all operations in an audit table

**3. 🧱 Functional Requirements**

**3.1 REST API Endpoints**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| POST | /employees | Create a new employee (XML input) |
| PUT | /employees/{id} | Update employee details |
| GET | /employees/{id} | Retrieve employee by ID |
| DELETE | /employees/{id} | Delete employee record |
| GET | /employees | List all employees |

**3.2 XML + XSD Validation**

* Incoming XML must conform to employee.xsd
* Validate fields like:
  + name: required
  + email: valid format
  + department: required
  + dateOfJoining: not in the future

**3.3 ActiveMQ Messaging**

* Publish events to employee.events queue on:
  + Create
  + Update
  + Delete
* Event payload:

json

{

"employeeId": 101,

"action": "CREATE",

"timestamp": "2025-10-22T04:45:00Z"

}

**3.4 JMS Listener**

* Listen to employee.events queue
* Log received events to the audit table

**3.5 Audit Logging**

* Create AuditLog table with:
  + id: primary key
  + employeeId
  + action: CREATE, UPDATE, DELETE
  + timestamp
  + source: REST or JMS

**4. 🧪 Non-Functional Requirements**

* Input validation using XSD and Java annotations
* Unit testing with JUnit and Mockito
* Use H2/MySQL for development
* Modular codebase with layered architecture

**5. 🧰 Technology Stack**

| **Layer** | **Technology** |
| --- | --- |
| Framework | Spring Boot |
| API | Spring Web |
| XML Binding | JAXB |
| Validation | javax.validation + XSD |
| Messaging | Spring JMS + ActiveMQ |
| Persistence | Spring Data JPA + MySQL/H2 |
| Testing | JUnit + Mockito |

**6. 📂 Sample XML Payload**

xml

<Employee>

<name>Priya Sharma</name>

<email>priya.sharma@example.com</email>

<department>Finance</department>

<dateOfJoining>2025-10-01</dateOfJoining>

</Employee>

**7. 📘 Sample XSD Schema (employee.xsd)**

xml

<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="Employee">

<xs:complexType>

<xs:sequence>

<xs:element name="name" type="xs:string"/>

<xs:element name="email" type="xs:string"/>

<xs:element name="department" type="xs:string"/>

<xs:element name="dateOfJoining" type="xs:date"/>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>

**8. 🗃️ Database Schema Design**

**8.1 employees Table**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| id | BIGINT | PRIMARY KEY, AUTO\_INCREMENT | Unique employee ID |
| name | VARCHAR(100) | NOT NULL | Full name |
| email | VARCHAR(150) | NOT NULL, UNIQUE | Email address |
| department | VARCHAR(100) | NOT NULL | Department name |
| date\_of\_joining | DATE | NOT NULL | Joining date |
| status | VARCHAR(20) | DEFAULT 'ACTIVE' | Employment status |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Record creation timestamp |
| updated\_at | TIMESTAMP | ON UPDATE CURRENT\_TIMESTAMP | Last update timestamp |

**8.2 audit\_logs Table**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| id | BIGINT | PRIMARY KEY, AUTO\_INCREMENT | Unique audit log ID |
| employee\_id | BIGINT | FOREIGN KEY → employees(id) | Related employee ID |
| action | VARCHAR(20) | NOT NULL | Action type (CREATE/UPDATE/DELETE) |
| timestamp | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Time of action |
| source | VARCHAR(20) | NOT NULL | Origin of action (REST/JMS) |
| message\_id | VARCHAR(100) | NULL | Optional message tracking ID |

**8.3 message\_queue\_log Table (Optional)**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| id | BIGINT | PRIMARY KEY, AUTO\_INCREMENT | Unique message log ID |
| employee\_id | BIGINT | FOREIGN KEY → employees(id) | Related employee ID |
| queue\_name | VARCHAR(100) | NOT NULL | Name of the queue |
| payload | TEXT | NOT NULL | Raw message content |
| direction | VARCHAR(10) | NOT NULL | SENT or RECEIVED |
| timestamp | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Time of message transaction |

**10. 🧠 Goals**

* Build REST APIs with XML support
* Validate XML using XSD /Java annotations
* Integrate ActiveMQ for messaging
* Implement event-driven architecture
* Log operations for auditing
* Write unit tests and document APIs