

**Documentation Technique - Application de Gestion de Stages**

**1. Introduction**

Cette documentation technique décrit l'application monolithique de gestion de stages développée pour faciliter le processus de stages entre les étudiants, les entreprises, les enseignants et les administrateurs.

**1.1. Objectif du projet**

L'application a pour objectif de digitaliser et simplifier l'ensemble du processus de gestion des stages, depuis la publication des offres par les entreprises jusqu'au suivi administratif des conventions de stage.

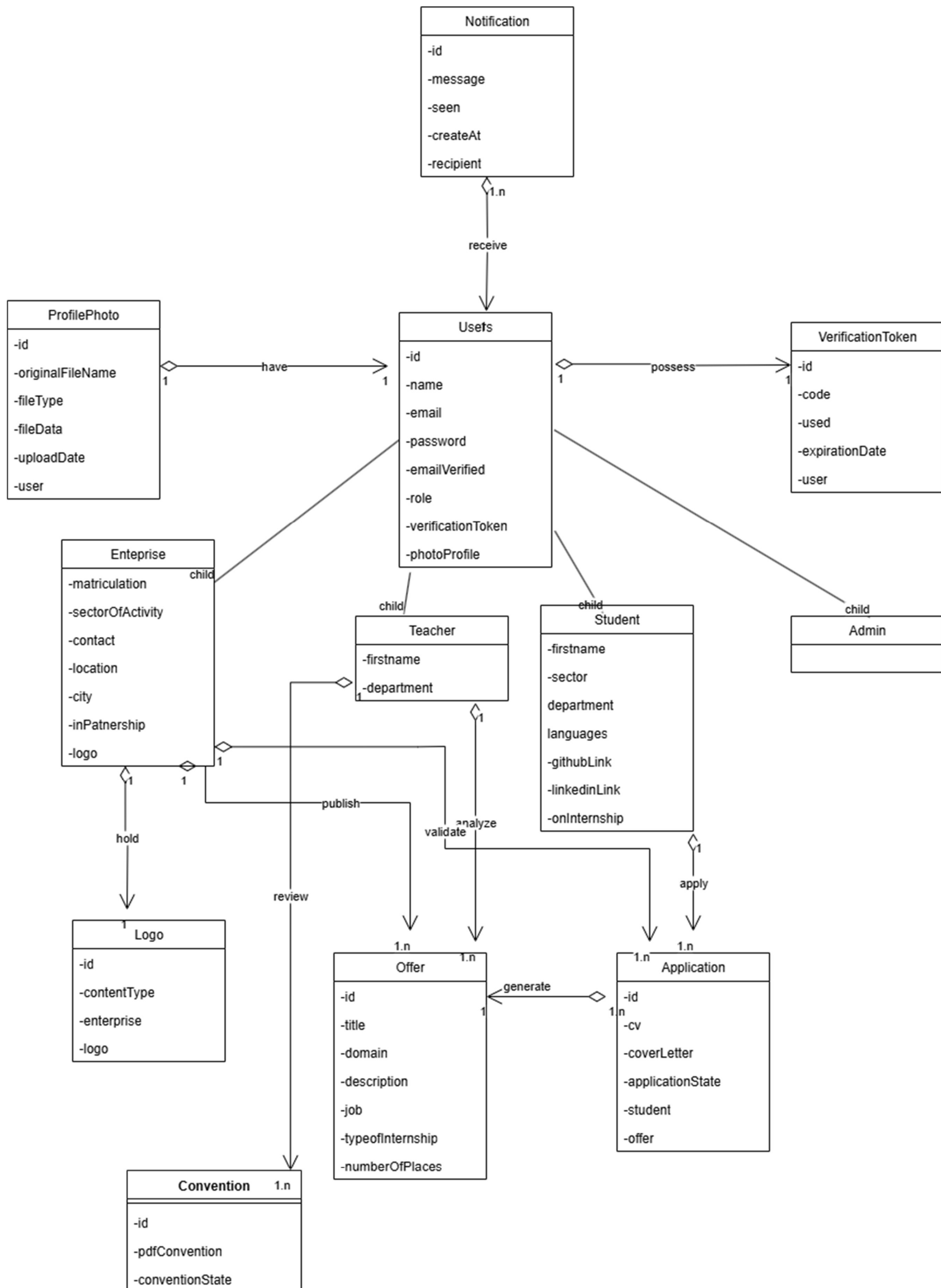
**1.2. Architecture**

L'application suit une architecture monolithique avec une base de données relationnelle. Elle est développée avec Spring Boot pour le backend, React pour le frontend, et PostgreSQL comme base de données.

**2. Diagramme de Classes**

Le système comprend les entités principales suivantes avec leurs relations :

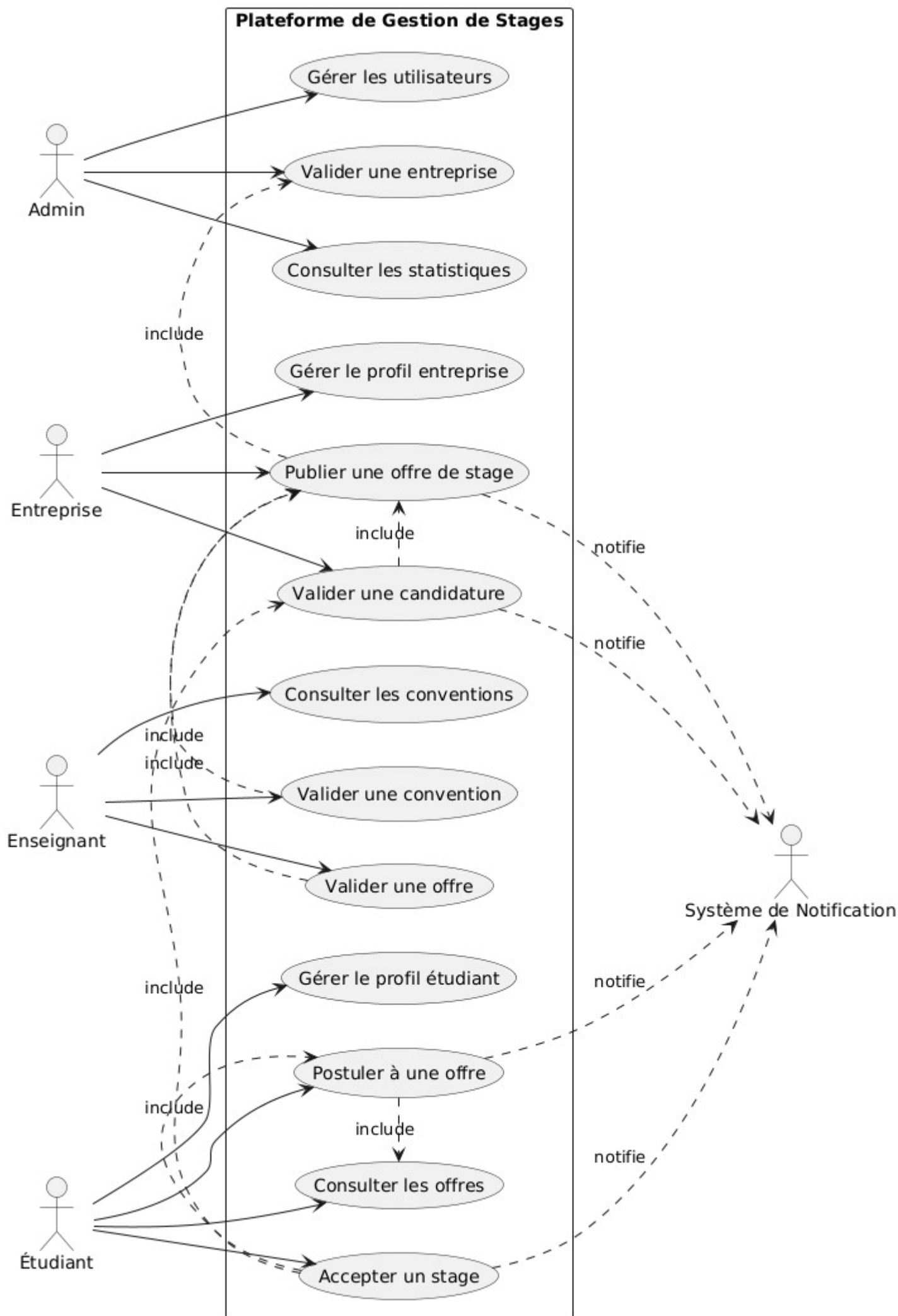
**3. Diagramme de Cas d'Utilisation**



### Diagramme de cas d'utilisation - Workflow principal

L'application comprend quatre acteurs principaux :

- **Admin** : Valide les entreprises, consulte les statistiques
- **Entreprise** : Publie des offres, valide les candidatures
- **Enseignant** : Valide les conventions et offres
- **Étudiant** : Consulte les offres, postule, accepte les stages



#### 4. Diagramme de Séquence - Workflow Principal

##### 4.1. Processus de publication et validation d'une offre

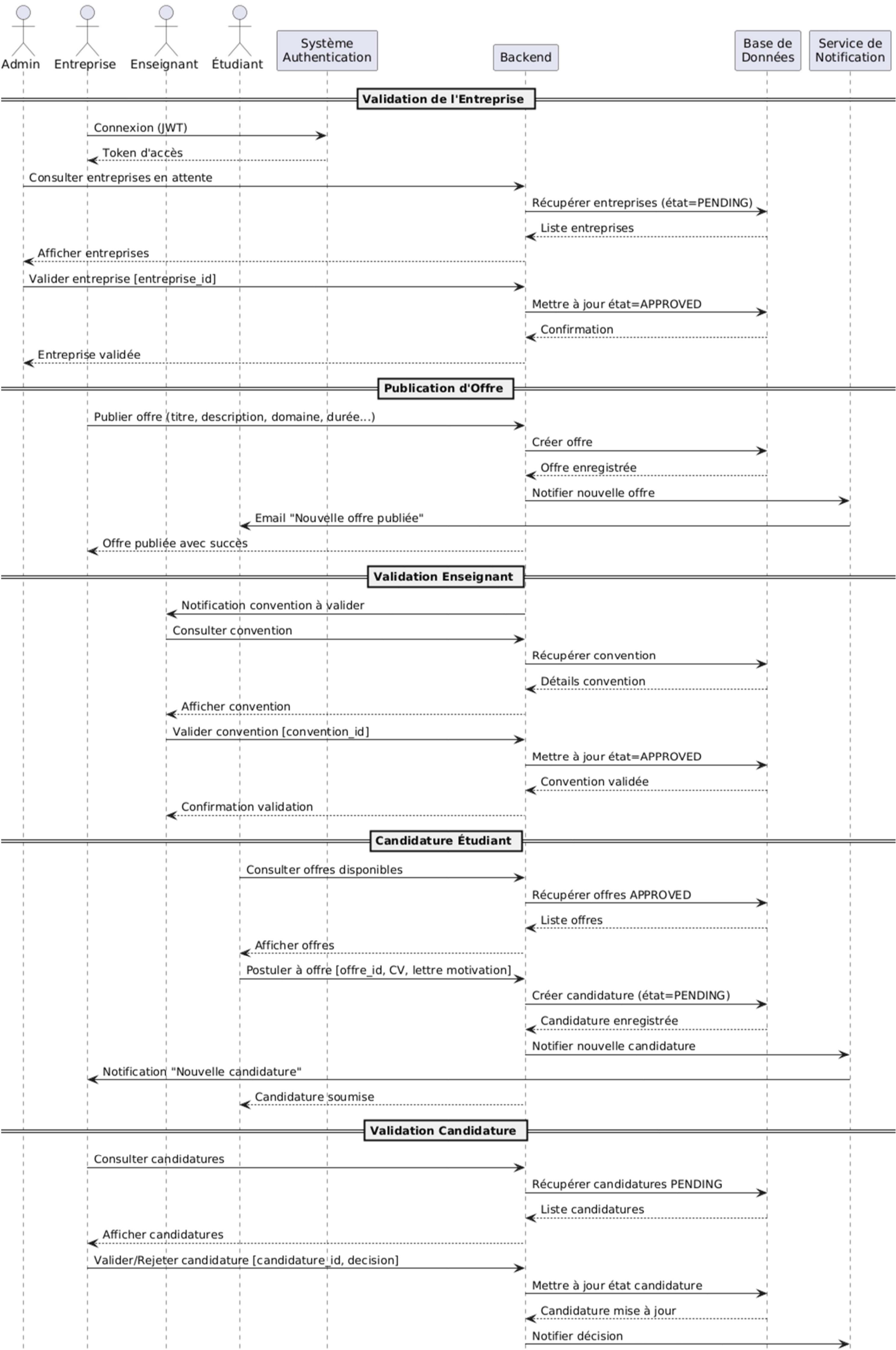
1. L'entreprise se connecte via JWT
2. Elle remplit le formulaire de publication d'offre
3. Le backend enregistre l'offre en base de données
4. Le système notifie tous les étudiants
5. L'enseignant valide la convention associée à l'offre
6. Le backend met à jour le statut de la convention

##### 4.2. Processus de candidature

1. L'étudiant consulte les offres disponibles

2. Il postule en soumettant CV et lettre de motivation
3. Le backend crée la candidature avec statut PENDING
4. L'entreprise est notifiée de la nouvelle candidature
5. L'entreprise valide ou rejette la candidature
6. L'étudiant est notifié de la décision
7. L'étudiant accepte finalement le stage

Workflow - Plateforme de Gestion de Stages



5. Architecture Technique

5.1. Stack Technologique

Composant	Technologie
Backend	Spring Boot, Spring Security, JPA/Hibernate
Frontend	React.js, Vite, Material-UI, figma
Base de données	PostgreSQL
Authentication	JWT (JSON Web Tokens)
Notifications	Service Email (SMTP), Notifications internes(webSocket)

5.2. Structure du Projet

```
src/
├── main/
│   ├── java/
│   │   └── com/
│   │       └── gestionstage/
│   │           ├── controller/    # Contrôleurs REST
│   │           ├── services/      # Logique métier
│   │           ├── repositories/  # Couche d'accès aux données
│   │           ├── entities/      # Entités JPA
│   │           ├── security/      # Configuration Spring Security
│   │           └── config/        # Configurations diverses
│   └── resources/
│       ├── application.properties # Configuration application
│       └── static/                # Assets statiques
└── frontend/
    ├── public/
    └── src/
        ├── components/           # Composants React
        ├── services/             # Services API
        └── styles/               # Feuilles de style
```

6. API Principale

6.1. Endpoints d'Authentification

Endpoint	Méthode	Description
localhost :8080/login	POST	Authentification utilisateur
localhost :8080/register	POST	Inscription nouveau utilisateur

6.2. Endpoints de Gestion des Offres

Endpoint	Description
/api/entreprise	Gère toutes les endpoints d’une entreprise

/api/teacher	Gère toutes endpoints d’un enseignant
/api/student	Gère toutes les endpoints d’un étudiant

7. Fonctionnalités Clés

7.1. Gestion des Utilisateurs

- Système d'authentification sécurisé avec JWT
- Profils multiples (Admin, Entreprise, Enseignant, Étudiant)
- Gestion des permissions et rôles

7.2. Workflow de Stages

- Publication et validation des offres de stage
- Système de candidatures en ligne
- Gestion numérique des conventions de stage
- Processus de validation multi-niveaux

7.3. Système de Notification

- Notifications email pour les événements importants
- Notifications internes dans l'application
- Paramétrage des préférences de notification

7.4. Reporting et Statistiques

- Tableaux de bord administratifs
- Export de données en format Excel
- Statistiques sur les stages et candidatures

8. Déploiement et Maintenance

8.1. Prérequis Système

- Java JDK 17 ou supérieur
- Node.js 14+ et pnpm
- PostgreSQL 12+
- Serveur SMTP pour les emails

8.2. Déploiement

L'application peut être déployée sur tout serveur supportant Java ou conteneurisée avec Docker.

8.3. Sécurité

- Validation des données côté serveur
- Protection contre les injections SQL
- Gestion sécurisée des tokens JWT