Rapport
Projet Application Web

AUGEREAU Robin - BENZERROUK Mathieu - CAMAIL Frédéric - AUBRY Jules

Département Sciences du Numérique - 2SN Réseaux
2023-2024

1 Ambition du projet

Le projet avait pour but de créer un site qui permettrait de mettre en relation des entreprises et des étudiants à travers des événements. Une entreprise pourrait y poster des événements tels que des visites, auxquels tout étudiant pourrait postuler. Une fois la demande réalisée, l'entreprise la valide (en fonction de la lettre de motivation de l'élève, sa classe ou encore son année) et indique le jour de l'événement si l'élève y est présent ou pas.
De l'autre côté, l'élève dispose d'une multitude d'événements dans plusieurs domaines (IA, Tech, etc.). Il peut les rechercher grâce à des filtres et choisir ainsi l'événement qui lui convient le mieux.

2 Structuration du code

2.1 Listes des entités

- *Avis :* permet à un utilisateur de donner un avis sur un événement. On peut ainsi voir, pour un événement, l'ensemble des avis sur ce dernier et, pour un utilisateur, tous ses avis. Un avis possède un titre, une note et un contenu (message).
- *Utilisateur :* permet à une personne de se connecter, s'inscrire, effectuer une demande pour un événement, envoyer des documents tels que la carte d'identité et donner des avis. Il existe néanmoins deux types d'utilisateurs : Élève (qui possède un INE, une classe) et Établissement. L'utilisateur possède en plus un nom, un mot de passe, un email, un token, et un téléphone.
- *Établissement :* permet la création d'un établissement, que ce soit une école ou une entreprise. Il possède un SIREN, une adresse, un nom, un booléen qui indique s'il s'agit d'une entreprise, une description et une image. Un membre d'entreprise peut ainsi créer un événement et lui attribuer un domaine.
- *Événement :* permet aux élèves de s'inscrire, aux entreprises de poster et de recevoir des avis sur un événement. Il possède un titre, une description, un créneau et une durée.
- *Document :* permet de caractériser un document. Il possède un nom et est rattaché à un utilisateur.
- *Domaine :* permet de caractériser le thème d'un événement et/ou d'une entreprise. Il possède un nom.

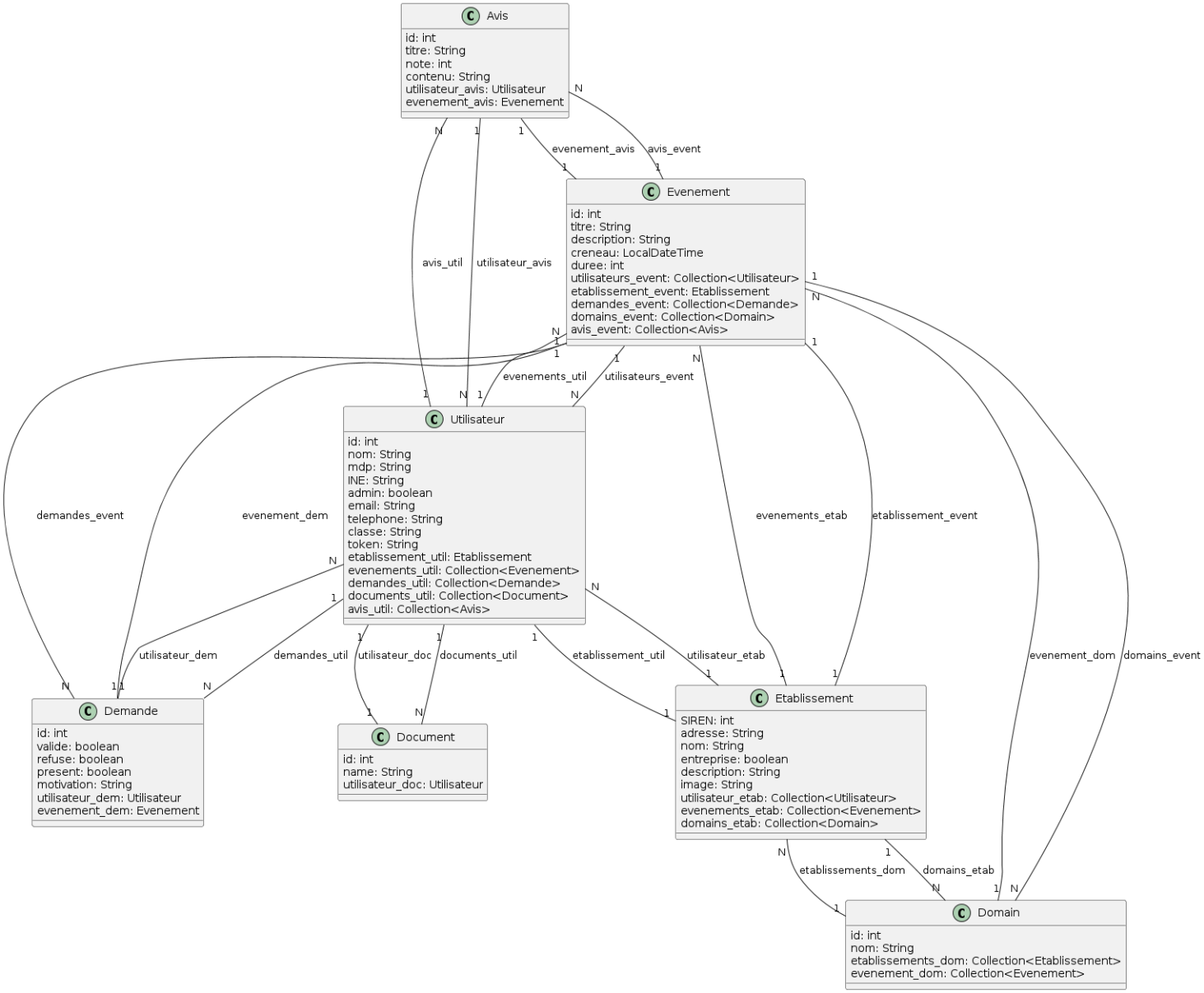

Figure 1: Diagramme UML du Back-End

3 Découverte de technologies

3.1 Côté back

1. De nouveaux outils
   - Wildfly
   - Maven pour les dépendances
   - Minio pour l'hébergement et la mise à disposition de fichiers
2. Difficulté rencontrées et solutions apportés
   - Hachage du Mot de Passe : Le mot de passe ne devait pas être transparant dans la BD. Pour cela nous avons utilisé **JbCrypt** afin que le mot de passe hacher avant d'être rentré dans la BD. Lors ce que l'utilisateur voulait ensuite se connecter, on hachage le mot de passe qu'il nous fournit à la connexion pour comparé au hachage stocké.
   - Envoie de Token par Cookies : L'utilisateur devait recevoir un token à la connexion et à l'enregistrement qu'il nous renverrait par cookies à chaque requête afin de s'identifier. Nous avons utilisé **BigInteger**, un librairie qui génère un token que nous avons simplement à renvoyé à l'utilisateur.
   - Communication par Json : Il fallait qu'on puisse envoyer tout type de variable au front. Pour cela nous sommes passé par des fichiers Json et utilisant un **GsonBuilde** qui transforme n'importe quel éléments du back en fichier

Json. Cependant quand on utilisait le builder pour transformer une entité on rencontrait un problème de bouclage sur les champs annotés @ManytoMany, @ManytoOne et @OnetoMany. Il a donc fallu les ignorer avec **@Expose** qui indique au builder quoi transformer. Les attributs marqués par les annotations @ManytoMany, @ManytoOne et @OnetoMany on dus être envoyés individuellement en les récupérant nous même de chaque entité par une fonction de la Facade.

3.2 Côté front

- Nuxt en tant que framework basé sur Vue.js
- TailwindCSS en tant que framework CSS
- shadcn-vue en tant que librairie CSS afin d'accélérer le développement

3.3 Côté développement et hébergement

- Développement sous Visual Studio Code
- Construction d'images de conteneur pour à la fois le développement et la "production"

4 Gestion de projet

- Tenue de réunions hebdomadaires
- Utilisation de Git et plus particulièrement de GitHub pour son interface simplifiée et ses issues
- Communication à travers des applications dédiées, avec une trace écrite des demandes d'implantation de fonctionnalités ou de correction de bugs à travers l'ouverture d'issues sur GitHub

5 Répartition des tâches

Tâches	Frédéric	Jules	Mathieu	Robin
Écriture des entités	X		X	
Résolution des issues écrites sur GitHub	X		X	
Écriture de la façade	X		X	
Écriture du servlet	X		X	
Écriture d'issues sur GitHub		X		X
Écriture de la vue utilisateur		X		
Écriture de la vue établissement				X
Création des environnements de développement et de déploiement				X

6 Avancement régulier

Semaine du 29 avril au 3 mai

Création des entités et leurs associations

- **Entités et Associations** (back/src/main/java/modele/)

Commencement des fonctionnalités dans la Facade

- **Facade** (back/src/main/java/vue/)

Commencement de la vue Utilisateur

- **Listing des évènements visible par les étudiant** (front/pages/etudiants/evenement/evenement.vue)
- **Page d'un événement avec le détail de l'entreprise** (front/pages/etudiants/evenement/[id].vue)
- **Page de compte d'un utilisateur** (front/pages/etudiants/[id].vue)

Vue Etablissement

- **Listing des événements créés par l'entreprise** (front/pages/etablissement/evenements.vue)
- **Page d'un évènement avec la liste des postulants** (front/pages/etablissement/evenement/[id].vue)
- **Demande d'un postulant pour un évènement** (front/pages/etablissement/demande/[id].vue)

Autres

- **Page d'inscription** (front/pages/auth/login.vue)
- **Page d'authentification** (front/pages/auth/signup.vue)

Semaine du 19 mai au 24 mai

Vue Etablissement

- **Création de la page de modification du compte d'un personnel d'un établissement** (front/pages/etablissement/dema
- **Création de la page pour créer et modifier un évènement** (front/pages/etablissement/evenement/modifier.vue)
- **Modification de la page d'un évènement avec la liste des postulants si l'évènement n'est pas passé, si l'évènement est en cours possibilité de faire l'appel, sinon le personnel voit des statistiques sur l'évènement** (front/pages/etablissement/evenement/[id].vue)
- Commencement de l'intégration de l'API Wildfly

Vue Etudiant

- **Création de la page de visualisation & modification du compte d'un étudiant** (front/pages/etudiants/[id].vue)
- **Création de la page qui permet de visualiser tous les évènements auxquels un étudiant a postulé** (front/pages/etudiants/mesEvenement.vue)
- **Ajout de filtres pour l'affichage de tous les événements** (front/pages/etudiants/evenements.vue)
- Commencement de l'intégration de l'API Wildfly

Facade

- Génération des tokens pour les utilisateurs après l'inscription et la connexion.
- Hachage du mot de passe à l'inscription avant de le rentrer dans la BD et comparaison avec le hachage du mot de passe rentré à la connexion.
- Création des fonctions nécessaires pour extraire les éléments de la BD pour le front.

Serveur

- Extraction du token des cookies et vérification pour les requêtes qui le nécessitent.
- Implémentation d'un builderJson pour envoyer les réponses de la facade sous format JSON pour le front.
- Création des conditions pour que le champ opérateur des requêtes utilise les bonnes fonctions de la facade.

BDD

- Modification des entités pour que le builderJson ne sélectionne que les champs voulus et ne boucle pas.

Semaine du 26 mai au 31 mai

Vue Etudiant

- **Création des avis pour commenter un évènement**
- **Ajout d'un icône pour suivre l'avancement de la demande**
- **Ajout de toutes les requêtes à la base de données**

Vue Etablissement

- **Finalisation des pages**
- **Ajout des fonctionnalités d'enregistrement et de lecture des pièces d'identité sur Minio**
- **Vérification de la correction de l'authentification**

Back-end

- Rajout de la plupart des fonctions renvoyant les attributs @ManytoOne, @ManytoMany et @OnetoMany des entités.
- Nouvelles fonctions : Savoir si un utilisateur est étudiant, des fonctions ne prenant que le token en paramètre, etc.

Environnement

- Construction des images de container
- Préparation du script de présentation