

Rapport projet JO2024

Partie UML

Sommaire

Introduction	3
Définition des cas d'utilisation	4
Editer les sessions	5
Editer les sites, disciplines et épreuves	5
Consulter les sessions, les sites olympiques, les disciplines et les épreuves	6
Authentifier un utilisateur	6
Diagrammes d'activité	8
Diagrammes de séquences	12
Diagrammes d'état-transition.....	15
Diagrammes de classe	18
Conclusion.....	19

Introduction

Les Jeux olympiques de Paris 2024 représentent un événement mondial de premier plan, et l'efficacité de leur organisation est cruciale pour garantir une expérience inoubliable aux athlètes et aux spectateurs. Dans ce contexte, le Comité international olympique a confié à notre équipe la mission passionnante de concevoir une application web dédiée à la gestion du planning des épreuves. Notre objectif principal est de développer une plateforme robuste et conviviale permettant aux organisateurs de coordonner efficacement les divers éléments liés aux Jeux olympiques, allant des sites olympiques aux disciplines, en passant par les sessions sportives. Pour atteindre cet objectif, nous utilisons une approche de développement basée sur l'Unified Modeling Language (UML) et le langage de programmation Java pour le côté serveur. Ce rapport détaillera l'architecture conceptuelle de notre application, mettant en avant les différentes entités, leurs relations et les interactions entre les acteurs. Nous présenterons également les règles de gestion cruciales qui garantiront la fluidité des opérations et la pertinence des informations affichées. Le projet est structuré en plusieurs modules, chacun correspondant à une fonctionnalité spécifique, allant de la gestion des sites olympiques à la visualisation des statistiques clé. Chaque aspect du système sera modélisé à l'aide de diagrammes UML appropriés, fournissant ainsi une représentation visuelle claire de notre conception.

Définition des cas d'utilisation

Nous avons décidé de définir trois use cases caractéristiques de projet permettant de décrire les principales fonctionnalités du site développé.

Les use cases retenues sont :

- Editer les sessions
- Editer les sessions, les sites, disciplines et épreuves (ces trois éléments ont un comportement similaire et sont donc considérés comme analogue dans nos schémas UML)
- Consulter les sites olympiques, les disciplines et les épreuves
- Authentifier un utilisateur

Par édition il est entendu les 4 opérations possibles sur un élément : création, modification, lecture et suppression.

Ces 4 UCs (use cases), sont rassemblés dans le diagramme de cas d'usages en figure 1. On y définit aussi les 4 profils possible pour un acteur du système. Le 1er profil est le simple visiteur qui a uniquement la possibilité de consulter les sites, disciplines et épreuves ainsi que les sessions. Les gestionnaires de sessions et gestionnaire administratifs ont les droits respectivement sur les sessions et les sites/disciplines/épreuves. Le profil administrateur est un héritage des deux profils précédents ayant donc tous les droits sur l'ensemble du site.

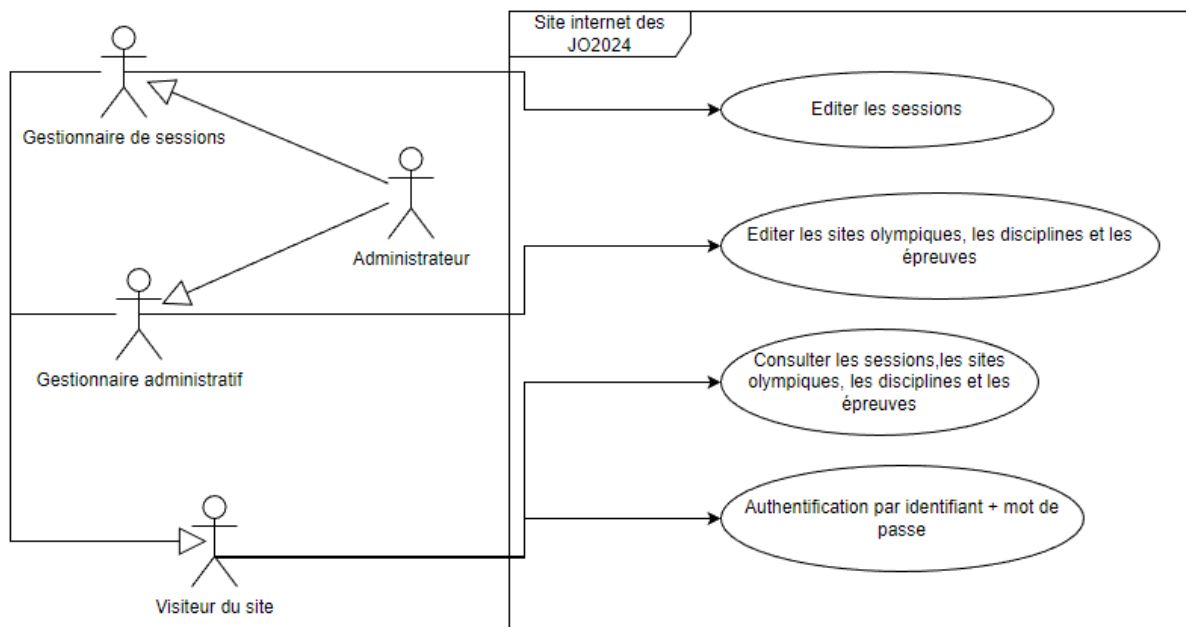


Figure 1: Diagramme d'usage des cas du projet

Editer les sessions

Description des cas d'utilisation

Le gestionnaire de sessions doit pouvoir effectuer toutes les opérations définies pour une session.

Déroulement de base :

1. L'utilisateur saisie les données sur la page dédié du site puis appuie sur le bouton d'envoi
2. Le serveur reçoit les données et les contrôle
3. Le serveur écrit en base de données les informations
4. Le serveur renvoie un message de confirmation à l'utilisateur
5. La page est rechargée avec les nouvelles données

Déroulement avec exceptions :

- A l'étape 2 en cas de superposition avec un session existante le serveur ne procède pas à l'écriture en base et envoie un message d'erreur à l'utilisateur précisant l'incident
- A l'étape 2 si la session existe déjà le serveur ne procède pas à l'écriture en base et envoie un message d'erreur à l'utilisateur précisant l'incident

Acteur principale : Le gestionnaire de session

Editer les sites, disciplines et épreuves

Description des cas d'utilisation

Le gestionnaire administratif doit pouvoir effectuer toutes les opérations définit pour les sites, disciplines et épreuves.

Déroulement de base :

1. L'utilisateur saisie les données sur la page dédié du site puis appuie sur le bouton d'envoi
2. Le serveur reçoit les données et les contrôle
3. Le serveur écrit en base de données les informations
4. Le serveur renvoie un message de confirmation à l'utilisateur
5. La page est rechargée avec les nouvelles données

Déroulement avec exceptions :

- A l'étape 2 si l'élément existe déjà le serveur ne procède pas à l'écriture en base et envoie un message d'erreur à l'utilisateur précisant l'incident

Acteur principale : Le gestionnaire administratif

Consulter les sessions, les sites olympiques, les disciplines et les épreuves

Description des cas d'utilisation

Le visiteur du site doit pouvoir consulter les différents éléments pour les JO2024.

Déroulement de base :

1. L'utilisateur accède à la page principale de l'application
2. L'application interroge la base de données
3. L'application récupère les données et les met en forme pour les afficher à l'utilisateur.

Déroulement avec exceptions :

- A l'étape 2 si l'application à un problème et ne peut récupérer les données stockées en base de données il affiche un message d'erreur.

Acteur principale : Le visiteur du site

Authentifier un utilisateur

Description des cas d'utilisation

Un utilisateur du site doit pouvoir s'authentifier pour accéder aux fonctionnalités de l'application associés à ses droits.

Déroulement de base :

1. L'utilisateur saisie ses identifiant et mot de passe puis appuie sur le bouton d'envoi
2. Le serveur reçoit les données
3. Le serveur interroge la base de données pour vérifier la combinaison donnée
4. Le serveur renvoie un message de validation à l'utilisateur
5. La page est rechargée et l'utilisateur est authentifier avec le profil correspondant

Déroulement avec exceptions :

- A l'étape 3 si la combinaison ne correspond pas à celle stocké en base de données l'application abandonne l'authentification et informe l'utilisateur que ses identifiants sont incorrects.

Acteur principale : Le visiteur du site

Diagrammes d'activité

Les diagrammes d'activité nous permettent de modéliser l'enchaînement d'action opéré par l'application pour réaliser l'use case défini. On notera dans les 3 premiers use cases la présence d'une étape authentification. Elle n'est pas détaillée, car correspond à l'action effectuée par le dernier use case et seront donc détaillées dans son propre diagramme.

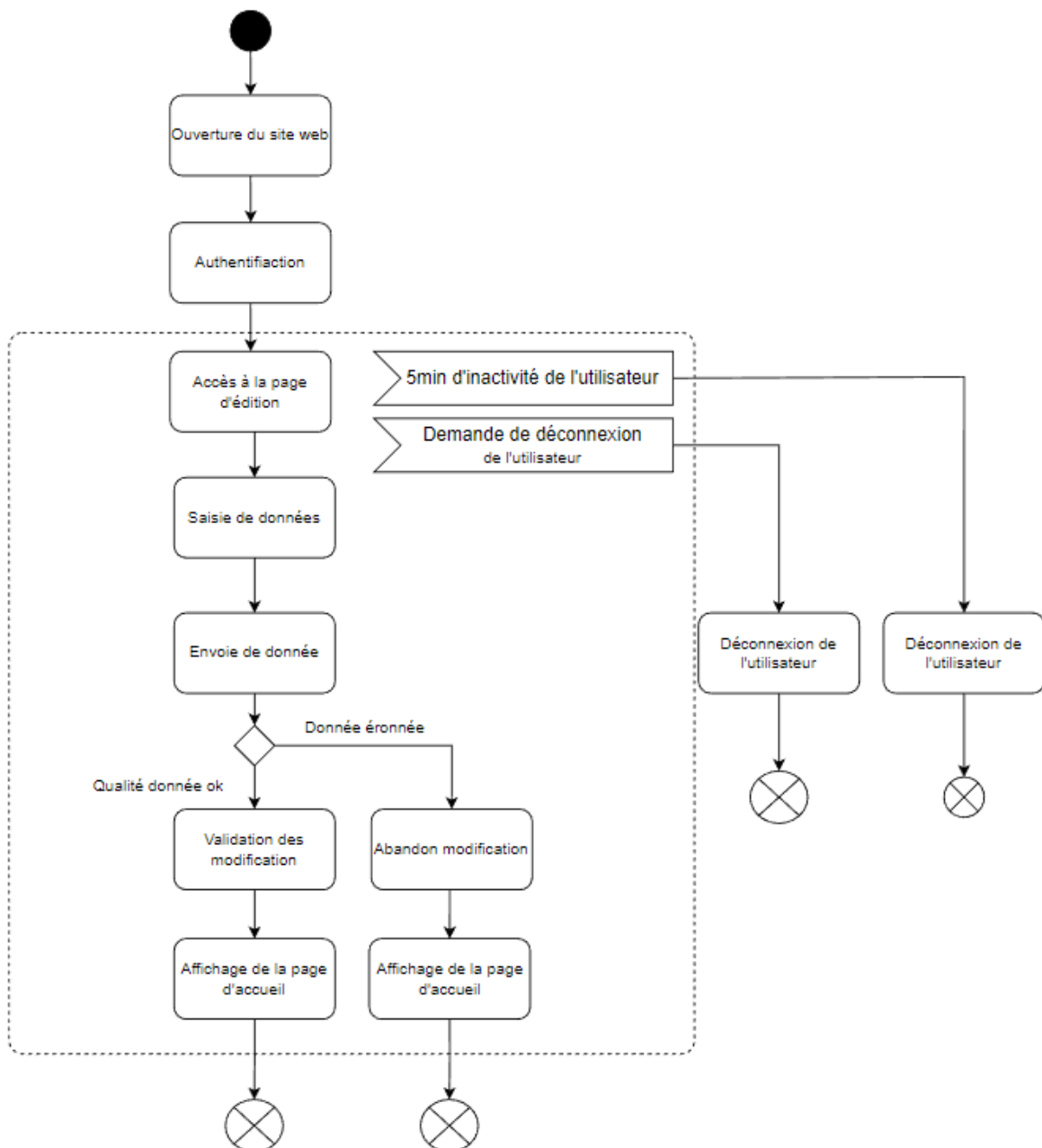


Figure 2: Diagramme d'activité pour l'édition de sites/disciplines/épreuves

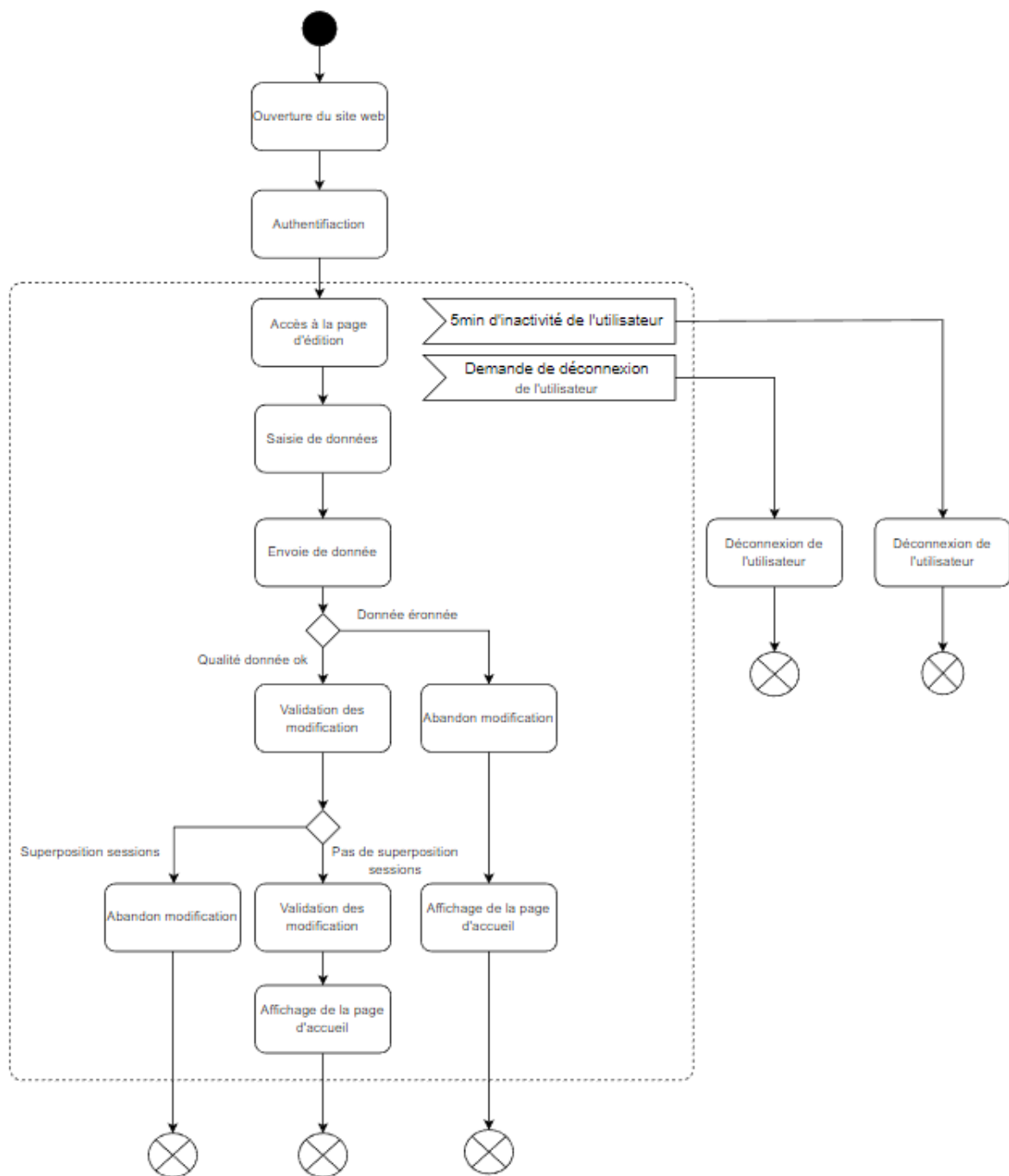


Figure 3: Diagramme d'activité de l'édition de session

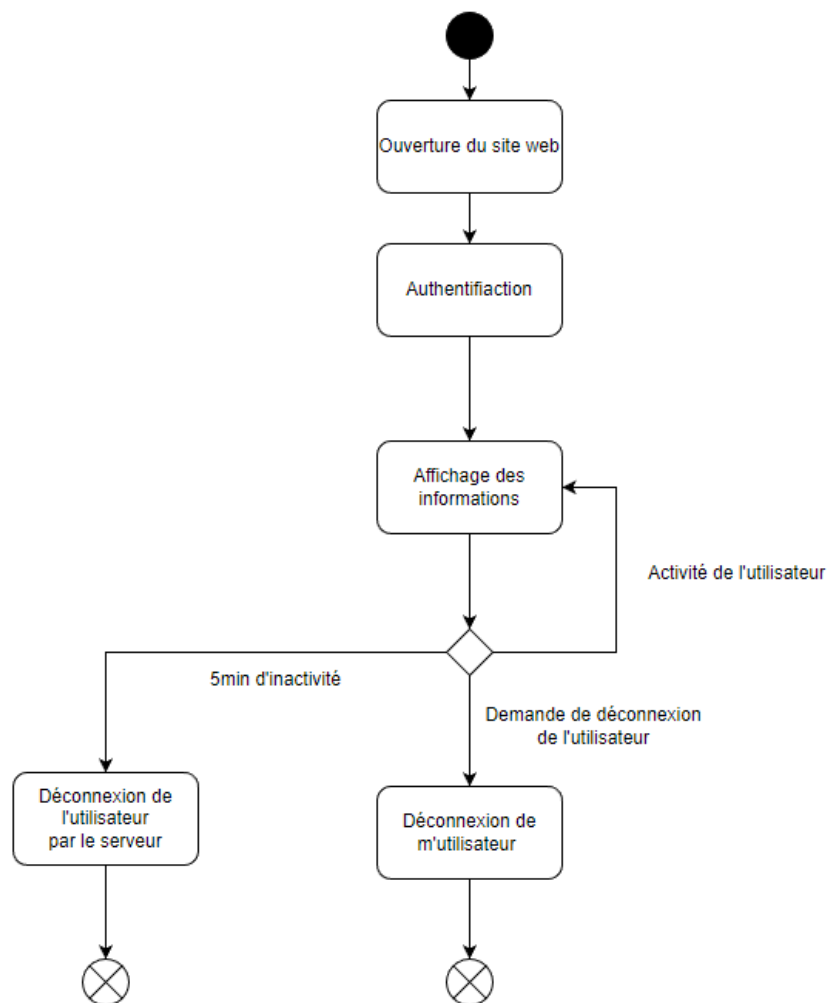


Figure 4: Diagramme d'activité de la consultation d'informations

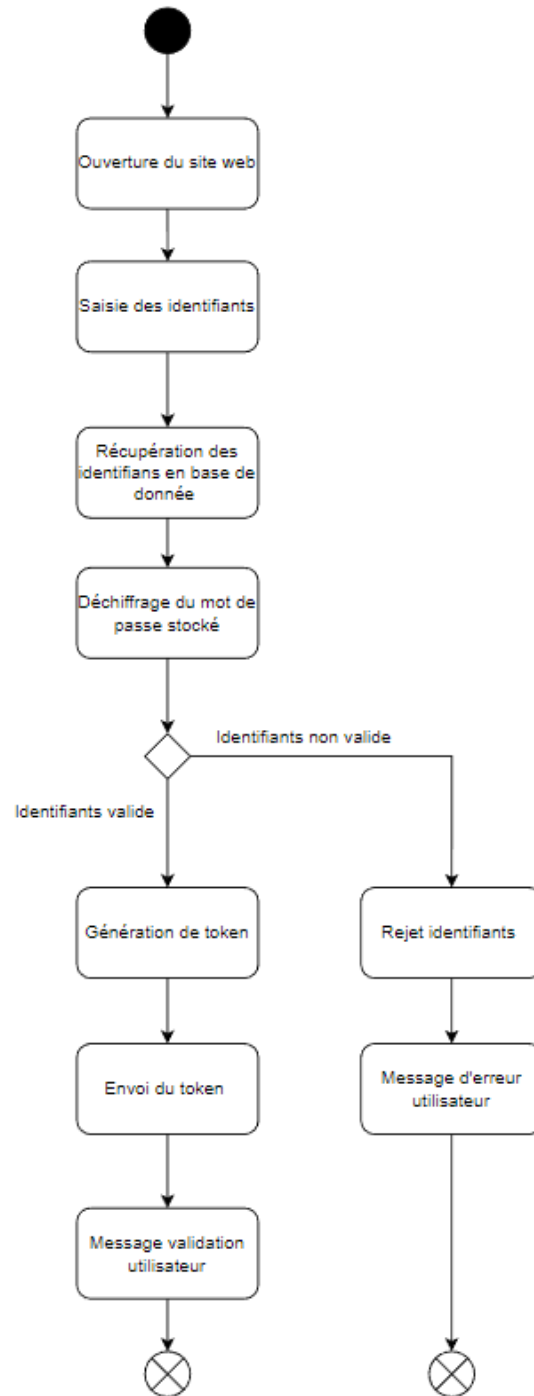


Figure 5: Diagramme d'activité associé à l'authentification

Diagrammes de séquences

Le diagramme de séquence établit la temporalité du déroulement d'un use case et fait partie des 1^{ers} diagrammes conçu pour notre projet. Ainsi il rend compte du fonctionnement global du système et de son interaction avec les acteurs de l'use case donnée. Nous avons ainsi suivi une approche boîte noire pour la modélisation du système. De plus le développement du projet s'appuie le framework spring.js et un diagramme en boîte ouverte complet forcerait à illustrer le comportement interne du framework ce qui présente peu d'intérêt.

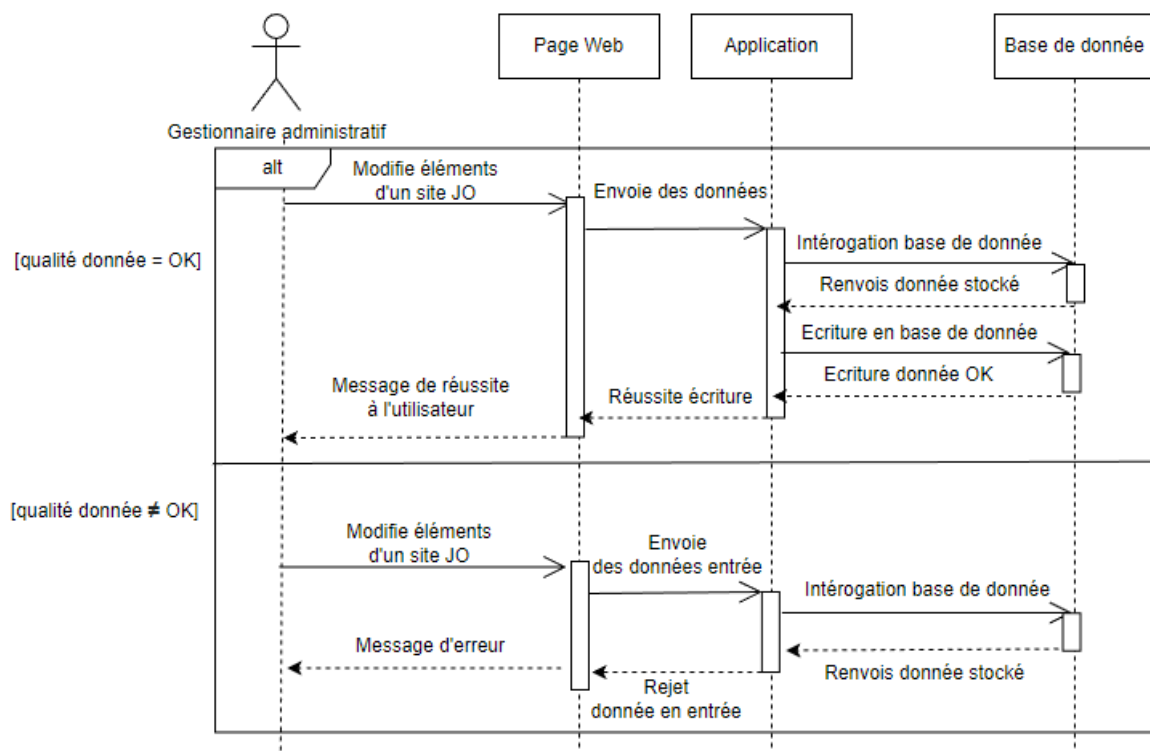


Figure 6: Diagramme de séquence pour l'édition d'un élément

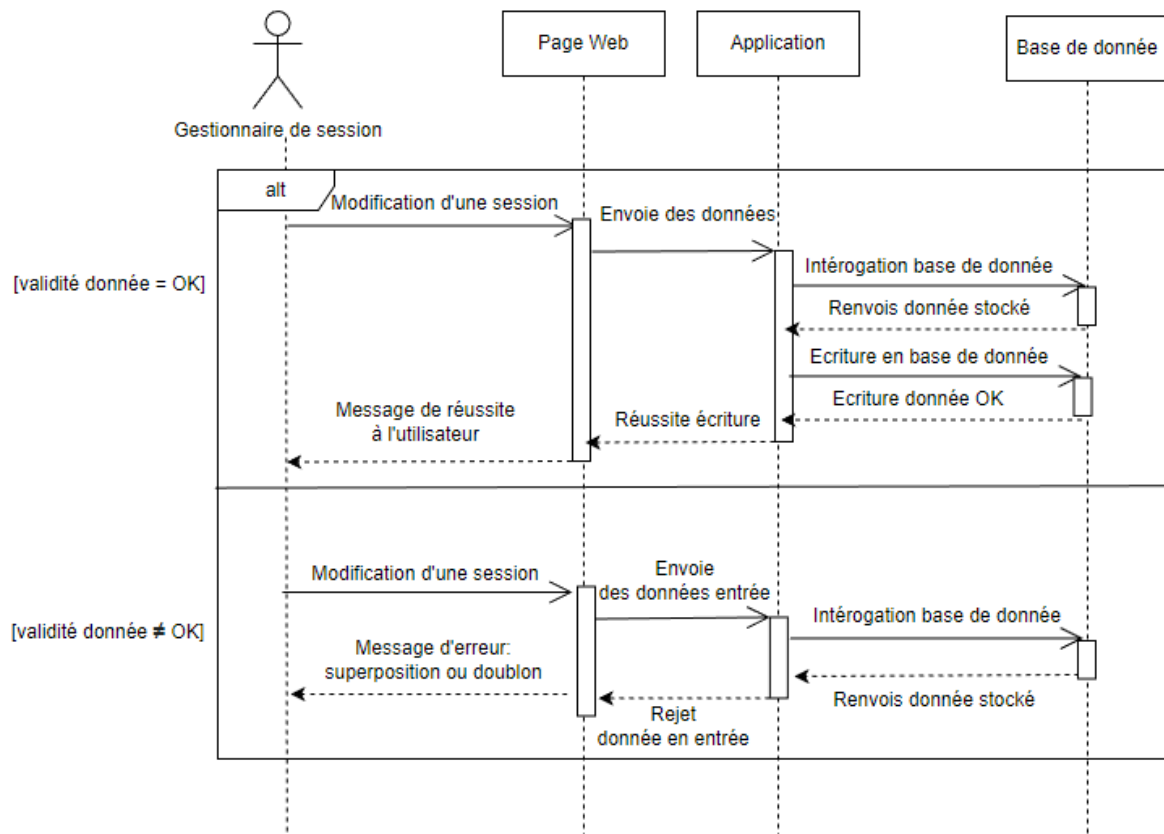


Figure 7: Diagramme de séquence de l'édition d'une session

Notre diagramme de séquence de l'édition de session ne distingue pas les deux cas d'exception décrit plus haut car les interactions entre les acteurs et l'application sont les mêmes, seul le contenu du message de réponse diffère.

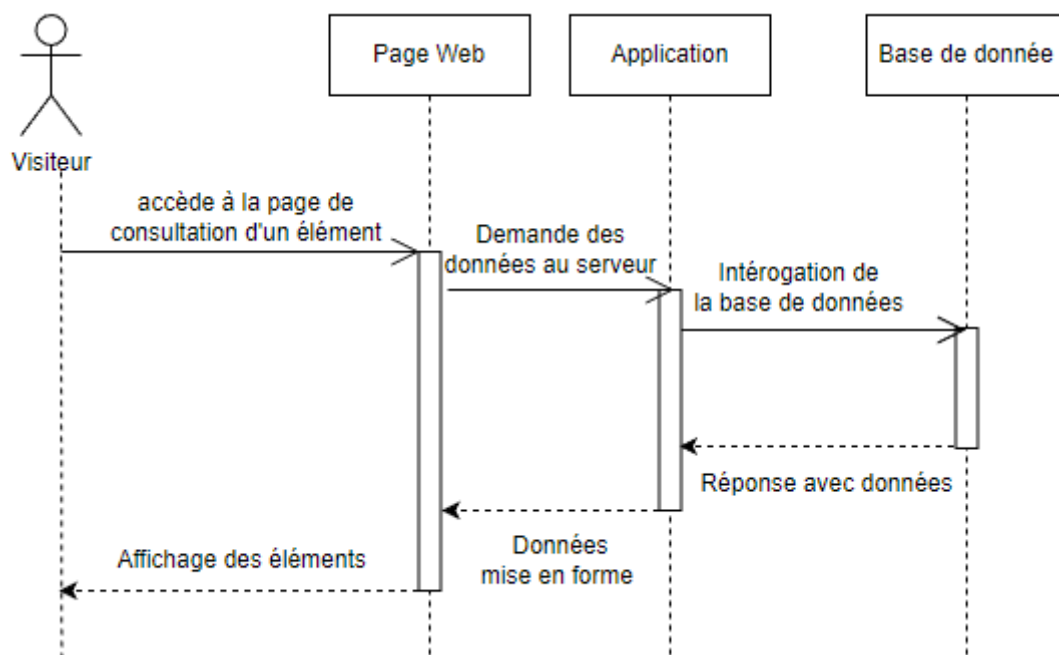


Figure 8: Diagramme de séquence de la consultation des éléments

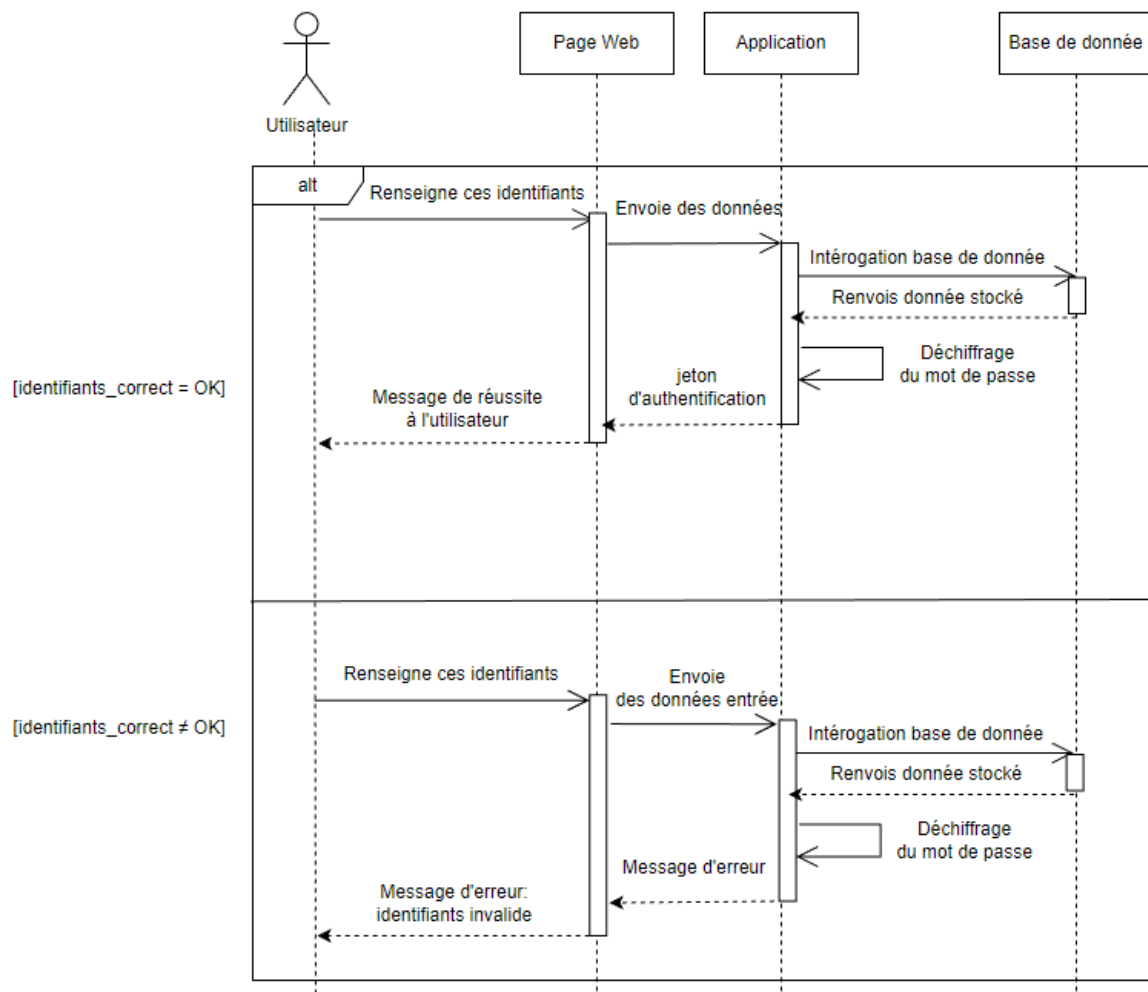


Figure 9: Diagramme de séquence du système d'authentification

On notera que l'authentification intervient aussi dans le processus d'édition des sessions et de sites/disciplines/épreuves. En effet l'identité de l'acteur est vérifiée via un token d'authentification dans la requête transmise par la page web à l'application. Cette tâche étant interne à l'application et « hors-périmètre » pour les autres use cases nous avons choisi de ne pas les faire apparaître.

Diagrammes d'état-transition

Les diagrammes d'états transitions nous permettent de décrire les états par lesquels passe l'application lors de l'exécution d'un use case. Nos diagrammes d'état-transition rentrent un peu plus dans les détails du fonctionnement de l'application décrivant son comportement à la réception de requêtes HTTP mais tout en essayant de conservé un degré d'abstraction élevé.

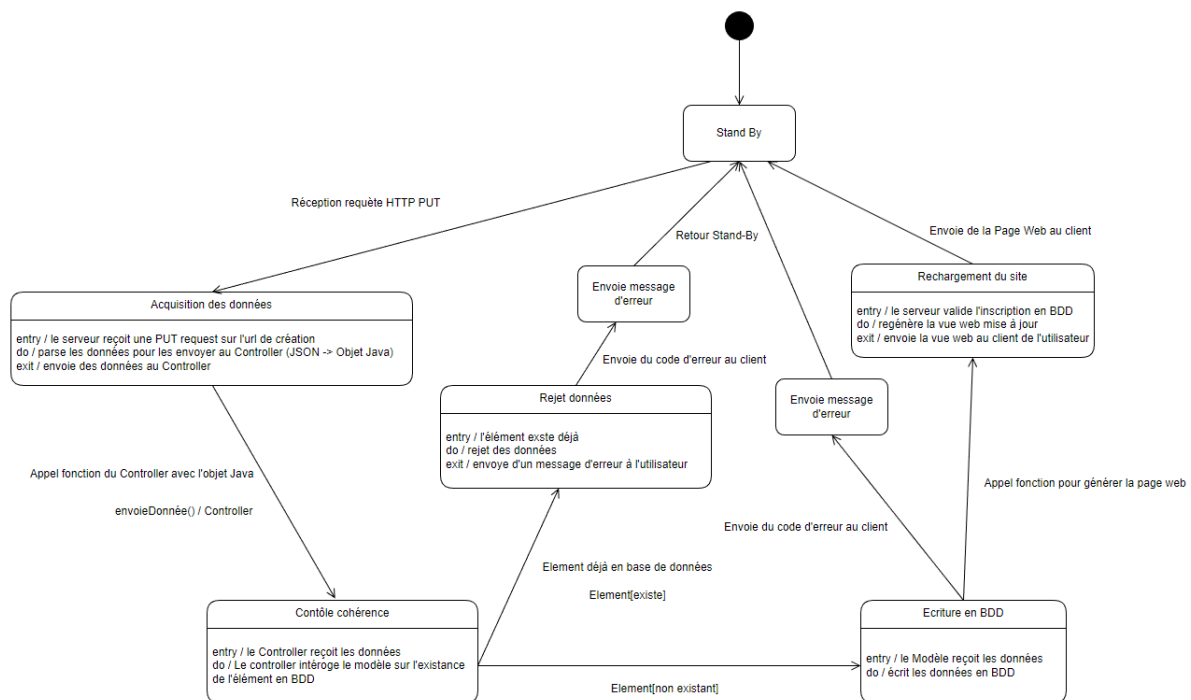


Figure 10: Diagramme d'état transition d'une création de site

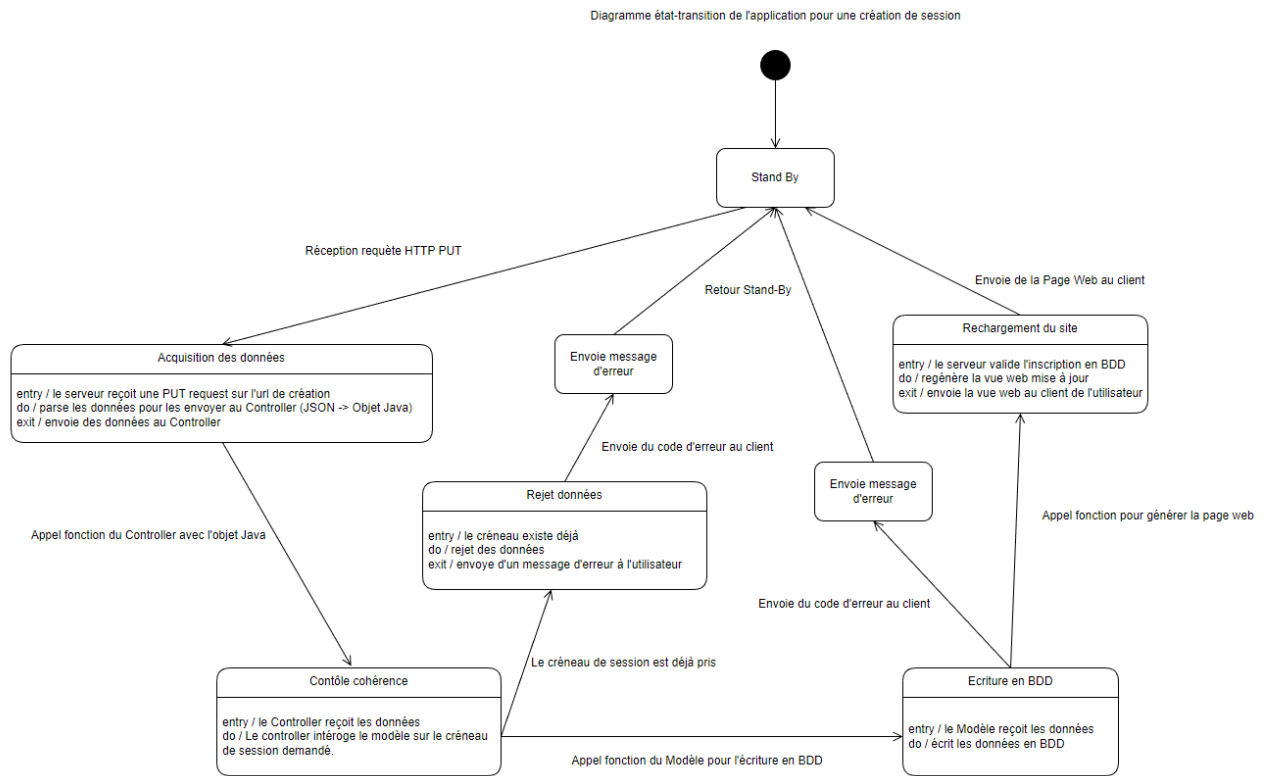


Figure 11: Diagramme d'état transition d'une création de session

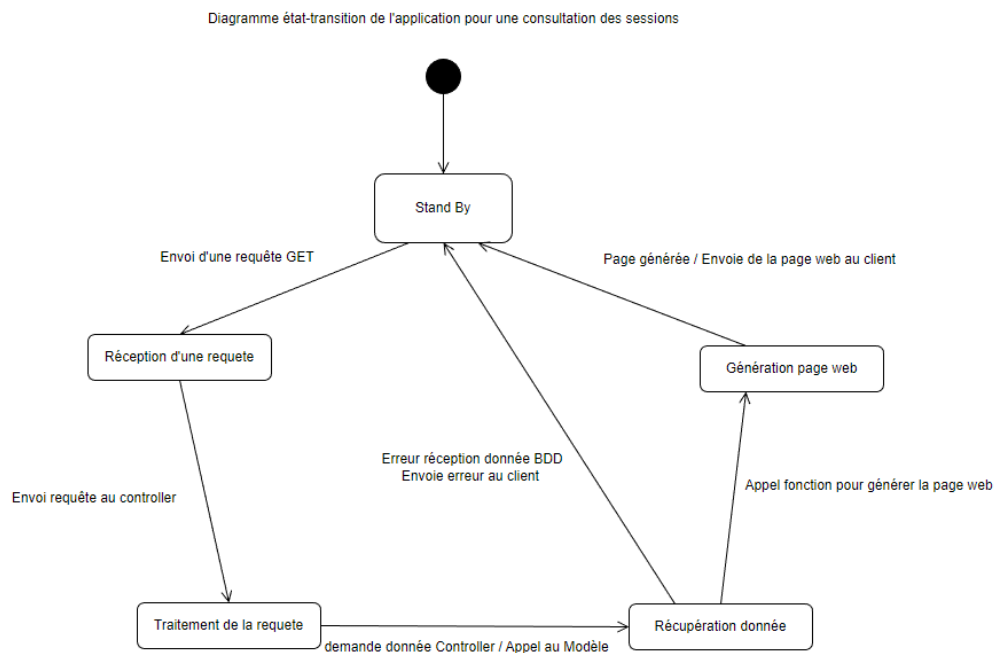


Figure 12: Diagramme d'état transition pour la consultation du site

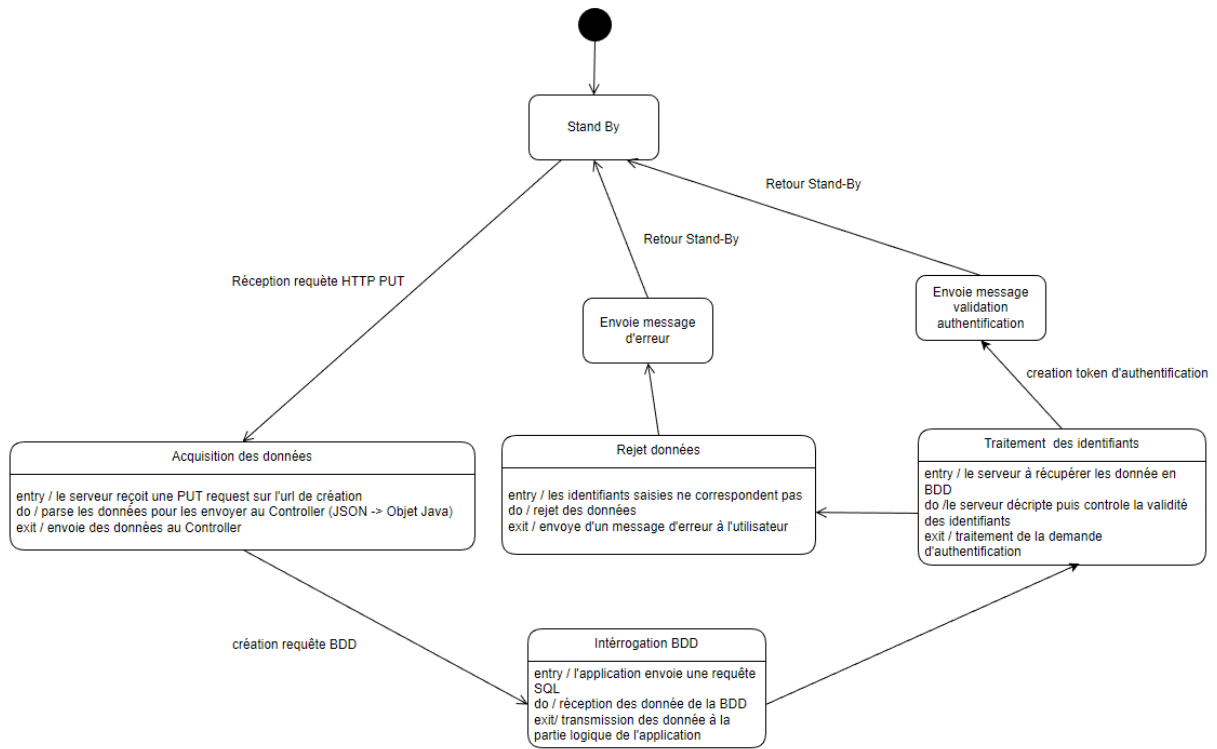


Figure 13: Diagramme d'état-transition de l'authentification

Diagrammes de classes

Le diagramme de classe nous permet d'identifier les différents objets que nous allons devoir coder ainsi que les relations qui les lient aux autres classes du projet. Il est à noter que les diagrammes de classes ici présentés ne correspondent pas strictement à ce qui a été développé. En effet, notre application repose sur Spring.js qui embarque plusieurs éléments nous permettant de ne pas avoir à réécrire certains objets.

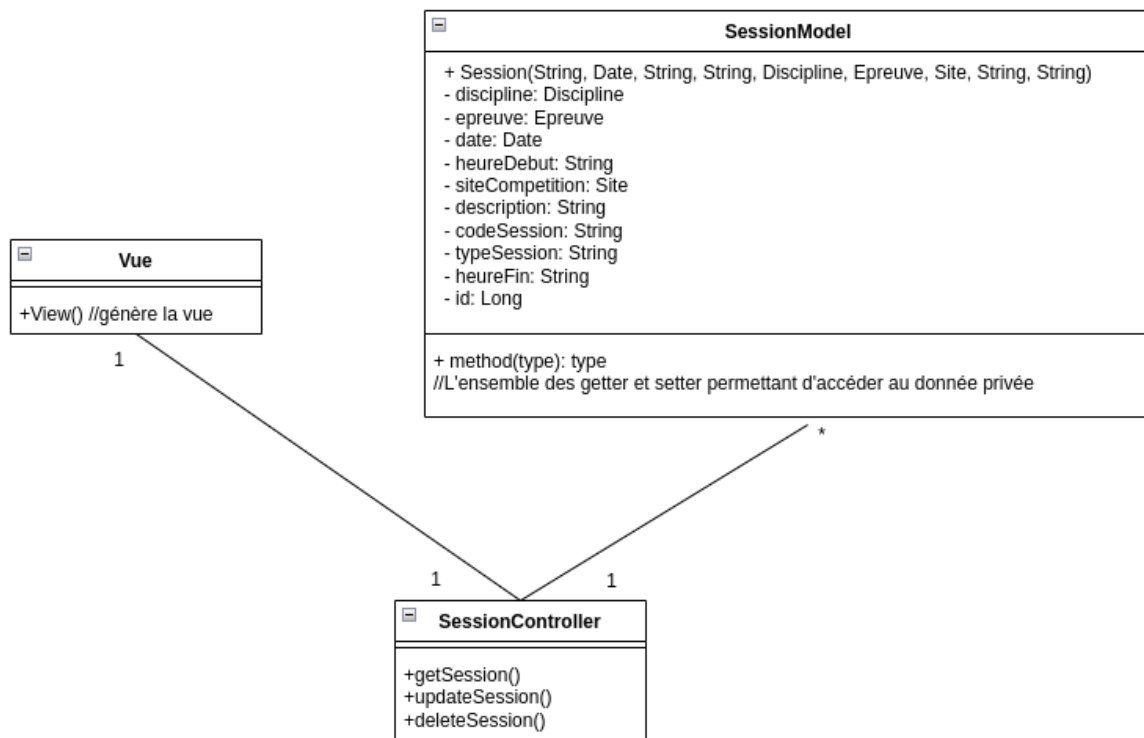


Figure 14: Diagramme de classe de l'édition de session

Par manque de temps, nous n'avons pas pu concevoir les autres diagrammes de classes qui auraient demandé plus d'élaboration et de réflexion.

Conclusion

À travers l'utilisation approfondie de l'Unified Modeling Language (UML), nous avons réussi à établir une architecture qui répond aux exigences du cahier de charges, nos diagrammes UML ont permis de détailler les principales caractéristiques du système et de structurer le développement de l'application final.

Enfin, ce projet nous a permis de constater l'intérêt que présente l'UML dans un projet aussi bien pour la conception que pour la veille technologique, ce dossier pouvant servir de base pour un dossier technique pour documenter l'application.