

5 Cadre général du projet

5.1 Contexte du projet

Il s'agit de la conception et le développement d'une application web de gestion des audits. Cette application doit disposer des fonctionnalités nécessaires à la gestion des auditeurs et des audits selon des normes reconnues.

Pour ce faire, nous avons commencé par une étude de l'existant afin de comprendre, en premier lieu, le fonctionnement global de ce type d'applications et de recenser, en second lieu les limites et lacunes des applications existantes.

5.2 Etude de l'existant

L'étude de l'existant est une étape préliminaire pour tout projet informatique, en vue de pallier les problèmes et les anomalies et d'apporter les corrections et les améliorations nécessaires aux procédures existantes. Dans le cadre de ce projet, nous avons choisi un principal module pour étudier ce qui existe déjà

5.2.1 Etude critique des plateformes concurrentes

Dans cette partie nous nous intéressons à l'étude d'une plateforme similaire à la nôtre. La plateforme concurrente choisie est:

-AudiTool

5.2.2 Analyse de la plateforme

AudiTool est un logiciel corporatif qui aide à contrôler tout le processus d'audit qui a pour but de remplacer la prise de note sous tableur, pour mettre en commun les notes d'audit et pour générer automatiquement les rapports d'audit.

Conception et réalisation d'une application De gestion des audits

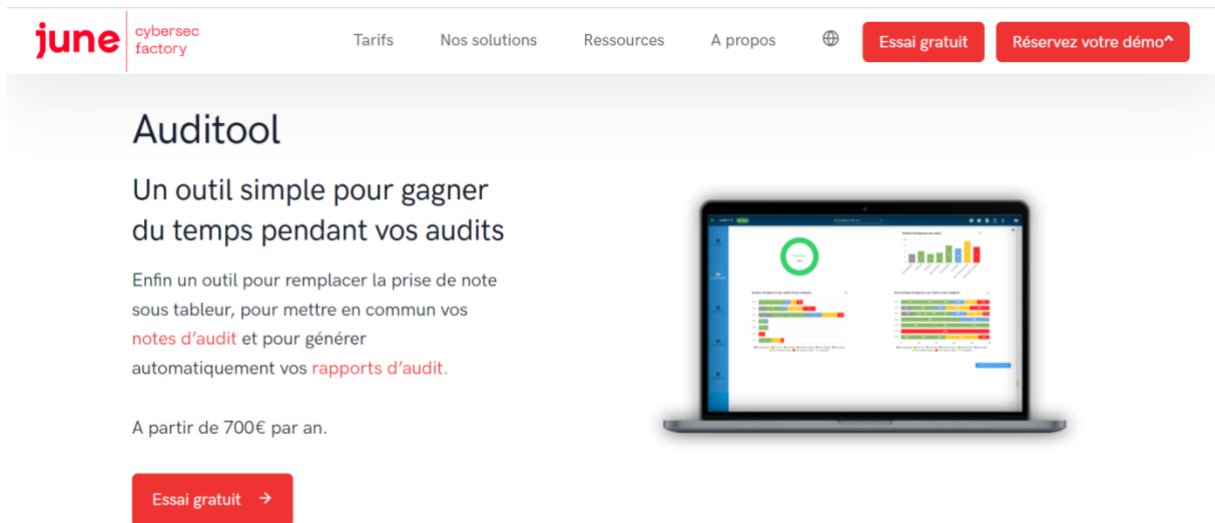


Figure 3 : Page d'accueil site AudiTool

Dans cette partie nous allons dégager les avantages et les inconvénients de la plateforme.

Tableau 4 : Points forts/faibles AudiTool

Points forts	Points faibles
Automatisation de tout le processus d'audit	Absence des statistiques automatiques
Automatisation de la génération des rapports d'audit	Pas de possibilité d'auditer d'autres exigences non incluses dans les standards
Ajout des standards via des importations Excel	Absence de la possibilité de gestion des permissions des utilisateurs
Accès hors ligne	Pas de gestion de traçabilité sur le site
	Pas d'extension mobile

5.3 Objectif de projet

Notre projet consiste à concevoir et à développer une plateforme qui doit répondre aux principaux objectifs suivants :

- Gain de temps : Optimisation du temps du cycle d'audit par la simplification de l'enregistrement des constats d'audit, le démarrage des plans d'actions correctives et l'attribution automatique des tâches
- Economie de l'argent : L'objectif principal de la plateforme est de permettre à une organisation d'auto-évaluer son système avant la demande de la certification dans un standard souhaité pour éviter le risque de perte d'argent

5.4 Spécification des besoins

Dans le but de réaliser une plateforme d'aide à l'audit, il est indispensable de recueillir et de formaliser les besoins et de l'ensemble des contraintes, puis l'estimation de la faisabilité de ces besoins.

Ces demandes du futur utilisateur seront exprimées au niveau de l'étape de l'analyse des besoins qui permet de mieux y répondre. Cette partie comporte une description détaillée de ces besoins à travers les diagrammes de cas d'utilisation qui permettent de décrire les spécifications du système à réaliser.

5.5 Etude de besoin

La plateforme doit satisfaire les besoins fonctionnels des différents acteurs et les besoins non fonctionnels qui identifient des contraintes internes et externes du système. Ces besoins seront représentés graphiquement par les diagrammes des cas d'utilisation

5.5.1 Identification des acteurs

Avant d'étudier les cas d'utilisation nous allons décrire les acteurs qui interagissent avec l'application en citant leurs rôles.

L'application peut être utilisée par, essentiellement, trois acteurs principaux:

- L'auditeur: c'est la personne qui utilise l'application pour effectuer des audits.

- **AdminOrganisation** : c'est la personne responsable de la définition du périmètre d'audit ainsi la gestion des auditeurs

L'application doit satisfaire les besoins fonctionnels des différents acteurs qui seront exécutés par le système et les besoins non fonctionnels qui identifient des contraintes internes et externes du système. Ces besoins seront représentés graphiquement par les diagrammes des cas d'utilisation.

5.5.2 Spécification des besoins fonctionnels

Les besoins fonctionnels représentent les actions que le système doit exécuter, il ne devient opérationnel que s'il les satisfait. Notre application doit couvrir principalement les besoins fonctionnels suivants selon le type de l'utilisateur:

- ✓ **S'authentifier** : La plateforme doit permettre à l'utilisateur de s'authentifier en 3 méthodes ; authentification normale en utilisant son login et mot de passe enregistrés sur la base de données, authentification via son compte Google ainsi qu'une authentification avec un serveur Active Directory. Cette fonctionnalité lui permet d'avoir la visibilité sur le contenu du site et d'avoir les accès sur les ressources qu'il contient
- ✓ **Gérer les organisations** : Ajouter, modifier, supprimer et lister les organisations.
- ✓ **Gérer les sites** : Ajouter, modifier, supprimer, lister les sites et associer un site a une organisation.
- ✓ **Gérer les processus** : Ajouter, modifier, supprimer, lister les processus et associer un processus a une organisation.
- ✓ **Gérer les standards** : Ajouter, modifier, supprimer, lister les standards.
- ✓ **Gérer les clauses** : Ajouter, modifier, supprimer, lister les clauses et ajouter une clause a un standard.
- ✓ **Gérer les systèmes de management** : Définir les systèmes de managements à auditer, modifier ou supprimer un système de management, ainsi qu'afficher la liste des systèmes de management.
- ✓ **Gérer les audits** : Ajouter, modifier, supprimer un audit, afficher la liste des audits, lancer un audit, cloner un audit, gérer les non conformités, gérer les corrections / actions correctives et gérer les plans d'actions.

5.5.3 Spécification des besoins non fonctionnels

Les besoins non fonctionnels décrivent toutes les contraintes techniques, ergonomiques et esthétiques auxquelles est soumis le site pour sa réalisation et pour son bon fonctionnement.

Les principaux besoins non fonctionnels de notre application sont :

- **L'ergonomie** : Les interfaces doivent être ergonomiques et conviviales. La manipulation des interfaces doit être simple et claire afin de s'adapter aux connaissances informatiques des utilisateurs.
- **La sécurité** : La plateforme doit garantir à l'utilisateur l'intégrité des données qui doivent garder leur forme et leur contenu original. En outre, elle doit protéger la confidentialité informatique de notre utilisateur. Ceci peut se faire entre autres par le moyen d'un mot de passe assurant le contrôle d'accès.
- **L'évolutivité** : La plateforme doit être ouverte à l'ajout de nouveaux modules et doit pouvoir répondre aux nouveaux besoins fonctionnels. Ceci nous donne la possibilité de maintenir notre solution et d'intégrer des nouvelles fonctionnalités.
- **Robustesse et maintenabilité** : Le système doit assurer une gestion complète des erreurs.

6 Modélisation du système

Le modèle conceptuel de données est une représentation statique du système d'information. Il a comme objectif de constituer une représentation claire et cohérente des données manipulées dans le système d'information.

6.1 Présentation d'UML



UML (Langage de modélisation unifié) comme son nom l'indique, un langage de modélisation qui permet de représenter et de communiquer les divers aspects d'un système d'information.

Il se définit comme un langage de modélisation graphique et textuelle, destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue.

L'UML supporte un riche ensemble d'éléments de notation graphique. Il décrit la notation pour les classes, les composants, les nœuds, les activités, les cas d'utilisations, les objets les états ainsi que la façon de modéliser les relations entre ces éléments. UML possède les caractéristiques suivantes :

- ✓ Une sémantique détaillée,
- ✓ Un mécanisme intégré d'extension,
- ✓ Un langage textuel associé utilisé pour la description des contraintes (Object Constraints Language)

6.2 Les diagrammes des cas d'utilisations

L'objectif fondamental de cette étape est d'identifier les principaux cas d'utilisation. Nous nous intéressons à la réalisation des diagrammes des cas d'utilisation. Ces diagrammes décrivent précisément les besoins du client final et spécifient le comportement attendu par le système à développer.

6.2.1 Diagramme de cas d'utilisation pour un auditeur

L'auditeur utilise l'application pour auditer toutes les clauses et les autres exigences définies dans un système de management.

La figure suivante décrit l'ensemble des cas d'utilisation lié à un auditeur.

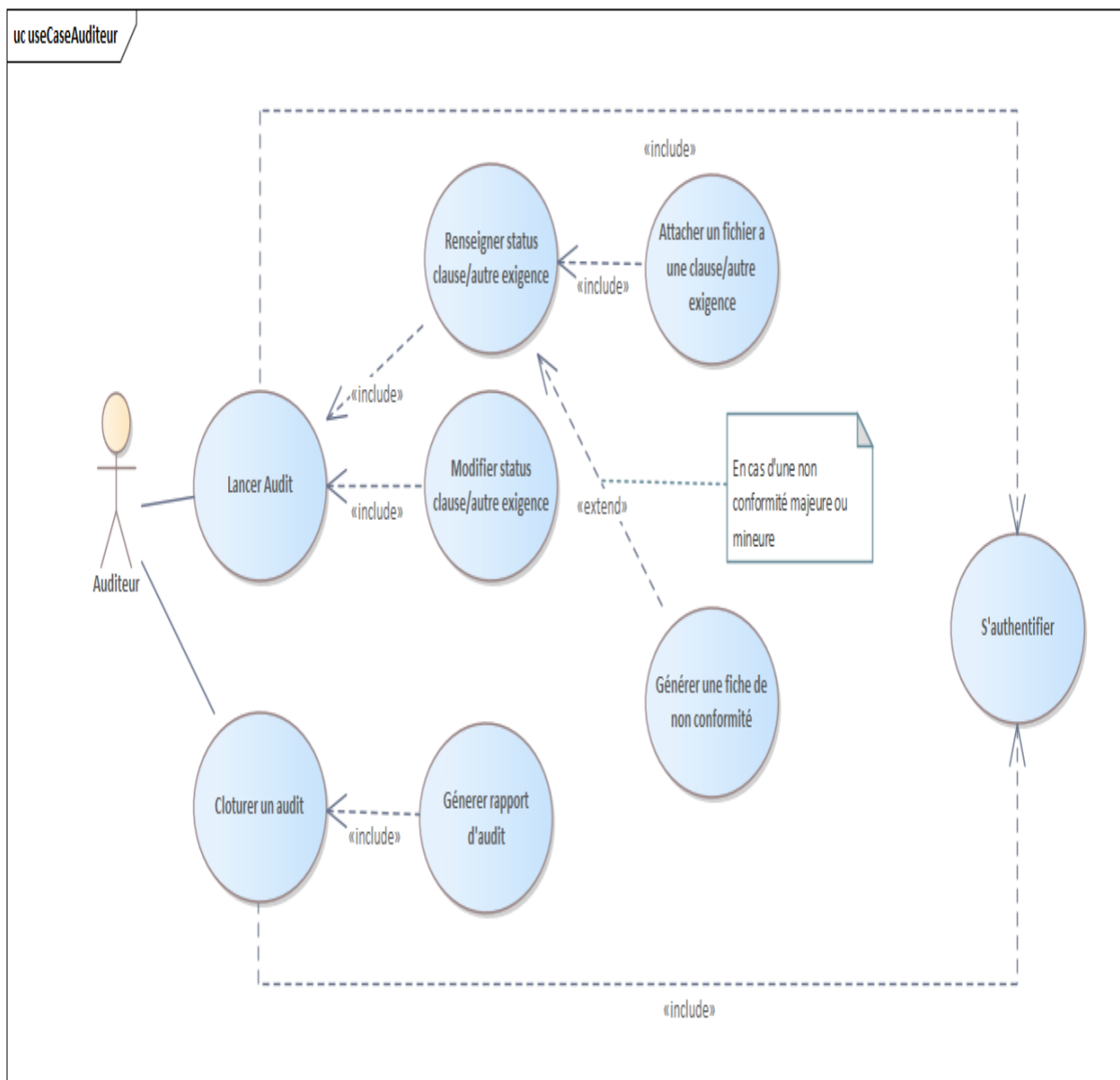


Figure 4 : diagramme de cas d'utilisation d'un auditeur

6.2.2 Diagramme de cas d'utilisation pour un adminOrganisation

L'adminOrganisation utilise l'application pour renseigner toutes les informations sur l'organisation , définir le périmètre d'audit , préparer l'audit pour l'auditeur ainsi qu'à consulter tous les résultats d'un audit et affecter des corrections/actions correctives aux personnes concernées.

Conception et réalisation d'une application De gestion des audits

La liste des figures suivantes décrit l'ensemble des cas d'utilisation lié à un adminOrganisation

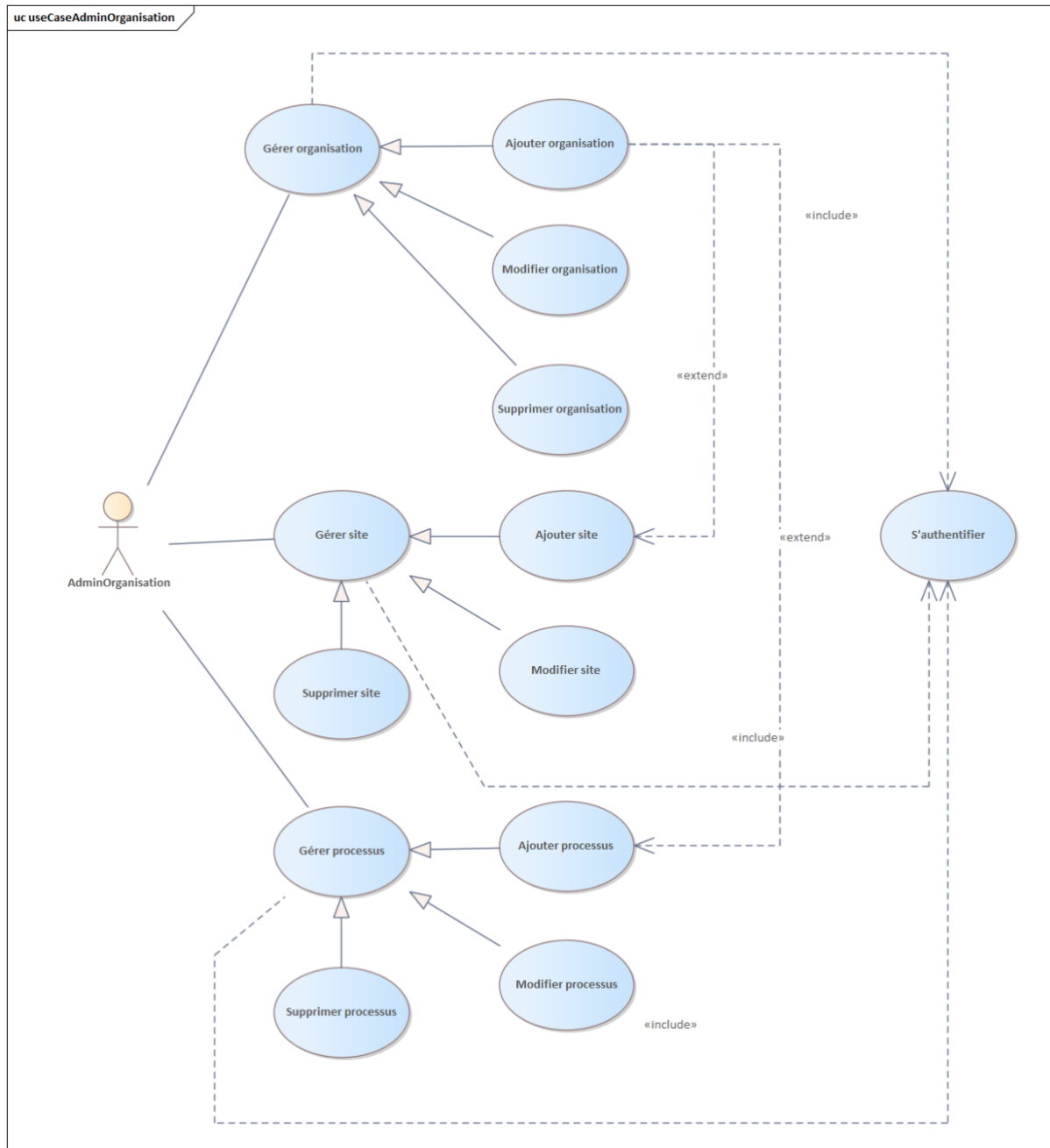


Figure 5 : partie 1 diagramme des cas d'utilisation adminOrganisation

Conception et réalisation d'une application De gestion des audits

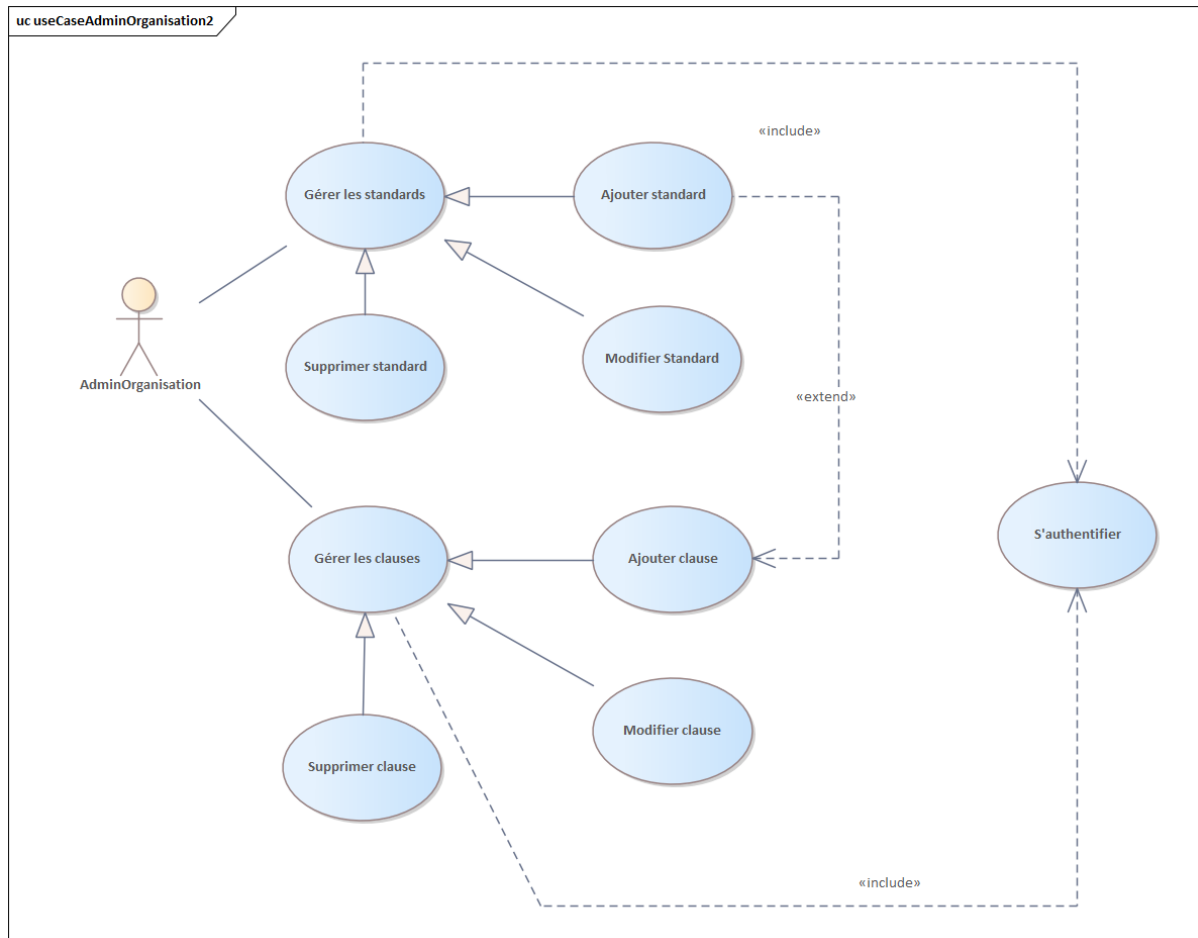


Figure 6 : partie 2 diagramme de cas d'utilisation adminOrganisation

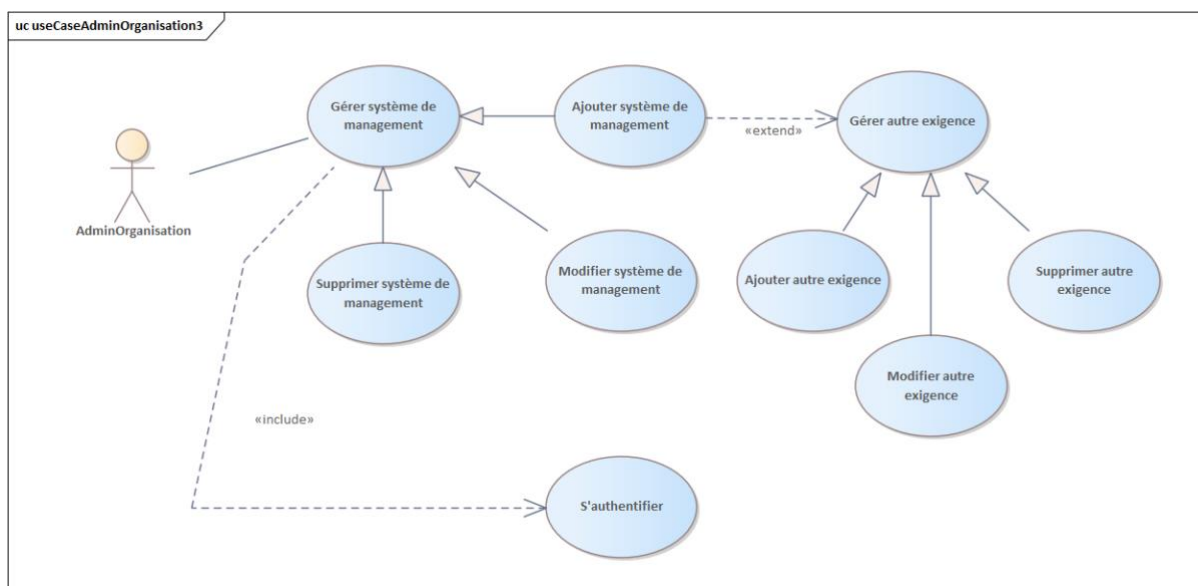


Figure 7 : partie 3 diagramme des cas d'utilisation AdminOrganisation

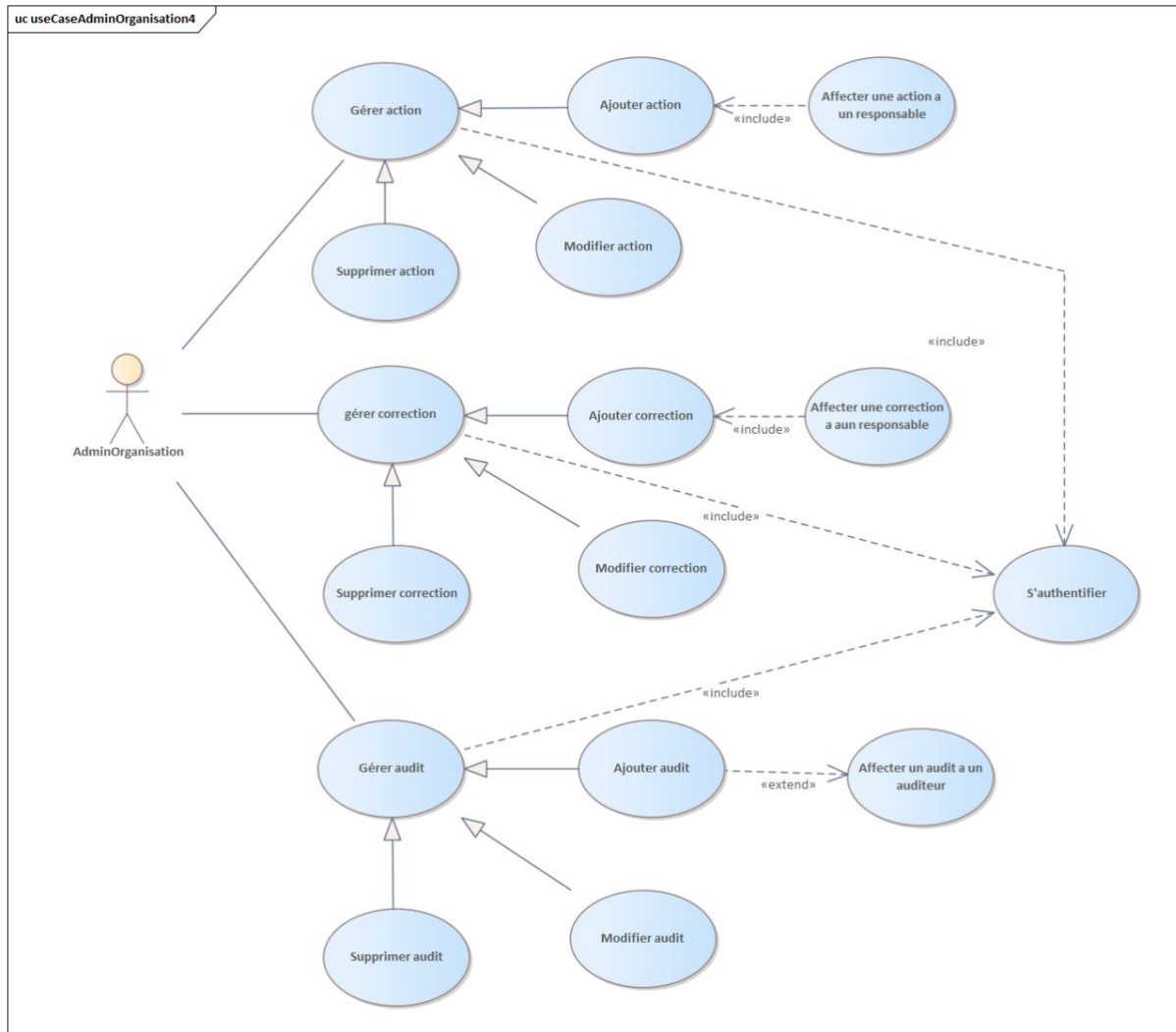


Figure 8 : partie 4 Diagramme des cas d'utilisation adminOrganisation

6.3 Diagramme de classe

Une classe représente la description abstraite d'un ensemble d'objet possédant les mêmes caractéristiques. Un objet est une entité à frontières bien définies possédant une identité et encapsulant un état et un comportement. C'est une instance ou une occurrence d'une classe. Une classe est constituée d'attributs dont les valeurs représentent l'état de l'objet et des méthodes qui sont les opérations applicables aux objets.

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci .Ce diagramme fait partie de la partie statique d'UML car il fait l'abstraction des aspects temporels et dynamiques.

Ce diagramme comporte l'ensemble des tables dont on a besoin pour le stockage des données

Conception et réalisation d'une application De gestion des audits

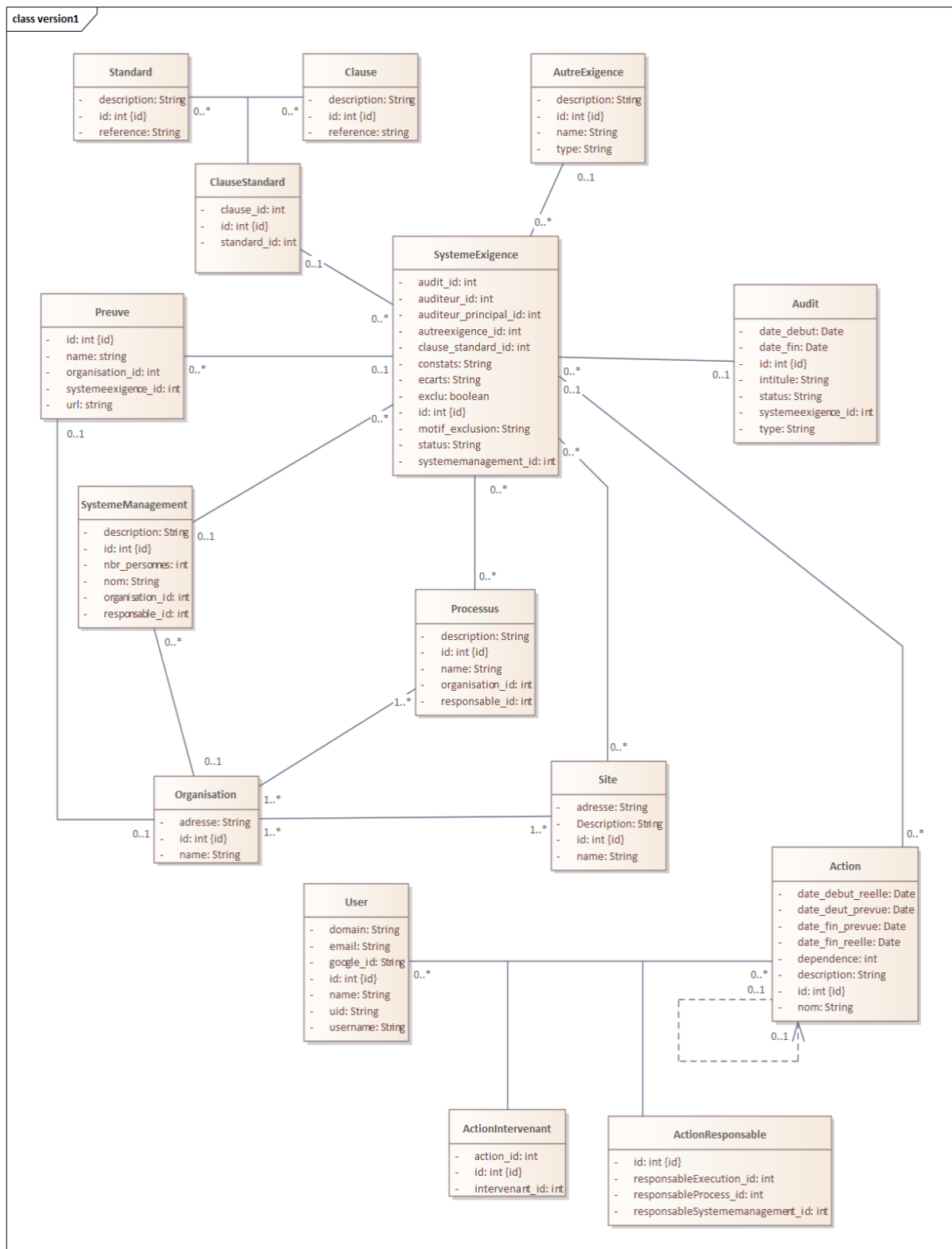


Figure 9 : Diagramme de classes