**Chapitre 2: Analyse et Conception**

**2.1 introduction:**

La réalisation d’une application Web doit être impérativement précédée par une méthodologie d’analyse et d’une conception, qui sont deux étapes primordiales pour mener à bien un projet informatique. Dans ce chapitre, nous commençons par une analyse des besoins fonctionnels et non fonctionnels et une identification des acteurs. Ensuite, nous passerons à la conception qui permet de décrire de manière non ambiguë le fonctionnement futur de l’application, en utilisant le langage de modélisation UML pour une bonne visualisation et représentation du projet.

**2.2 Spécification et analyse des besoins:**

Cette étape a pour objectif de présenter les besoins auxquels doit répondre notre application Web.

**2.2.1 Les besoins fonctionnels:**

* Créer, modifier et rechercher un compte utilisateur.
* Activer ou désactiver un compte utilisateur.
* Se notifier des nouvelles tâches.
* Crée, consulter une tâche.
* Cree, consulter un ticket.
* Crée, consulter un projet.
* Valider tickets.
* Valider taches.

**2.2.2 Les besoins non-fonctionnels:**

* Fournir à l’utilisateur une interface simple, conviviale, intuitive et ergonomique.
* Faciliter le travail collaboratif.
* Permettre un accès rapide à l’information.

**2.2.3 Identification des acteurs:**

Un acteur représente une entité jouant un rôle défini par une ou plusieurs fonctions, cette entité est externe au système mais elle est en interaction avec celui-ci

* **Administrateur :** c’est la personne qui s’occupe de la gestion des comptes de l’entreprise. Il veille à attribuer des comptes aux utilisateurs, tout en assurant une sécurité maximale.
* **Supérieur :** c’est celui qui a le droit de créer des projets (affecter plusieurs tâches à plusieurs utilisateurs) et visualiser toutes les tâches et projets et leurs états d'avancement selon sa direction/département.
* **Département help-desk :** c’est celui qui a le droit de créer des tickets aux utilisateurs (équipe de support) et de consulter leurs états d'avancement.
* **Opérateur :** c’est celui qui a le droit de créer des tâches aux utilisateurs (informaticien) et de consulter leurs états d'avancement.
* **Utilisateur :** C'est celui qui reçoit les tâches à effectuer et doit les valider une fois terminer et peut consulter l'état d'avancement des projets dont il fait partie.

**2.3 Conception:**

**2.3.1 Choix d’UML comme outil de modélisation:**

UML est l’approche moderne des logiciels de modélisation et de documentation la plus populaire. Nous avons utilisé cet outil pour ses nombreux avantages :

* Il permet de documenter très clairement les besoins exprimés par les utilisateurs.
* Il permet d’établir une structuration cohérente des fonctionnalités et des données.
* Il permet une compréhension précise des concepts objets du système que l'on veut modéliser.

**2.3.2 Diagramme de cas d’utilisation:**

Un cas d’utilisation (Use Case) représente un ensemble de séquences d’actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier. Chaque cas d’utilisation spécifie un comportement attendu du système considéré comme un tout, sans imposer le mode de réalisation de ce comportement. Il permet de décrire ce que le futur système devra faire, sans spécifier comment il le fera.

Nous commencerons par présenter le diagramme de cas d’utilisation global (Figure 4) afin de donner une vision globale du comportement fonctionnel de notre application, puis nous détaillerons les autres diagrammes de cas d’utilisation du système.



Figure 1 : Diagramme de cas d’utilisation global

Le supérieur peut consulter les taches et en crée quel que soit leur type (tache, projet). Et peut aussi modifier son profil et mot de passe. La (figure 5) représente le diagramme de cas d’utilisation de ce rôle.

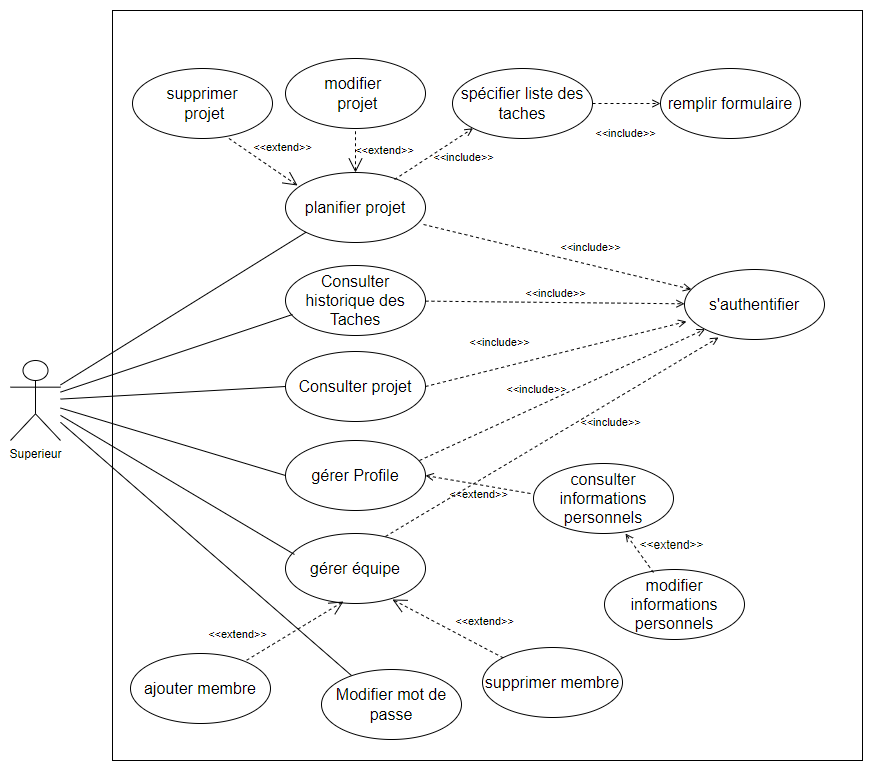


Figure 2 : Diagramme de cas d’utilisation du supérieur

Le help-desk peut crée des tickets et consulter l’Etat d’avancement en temps réel, peut aussi modifier ou supprimer un ticket si besoin. La (figure 6) représente le diagramme de cas d’utilisation de ce rôle.

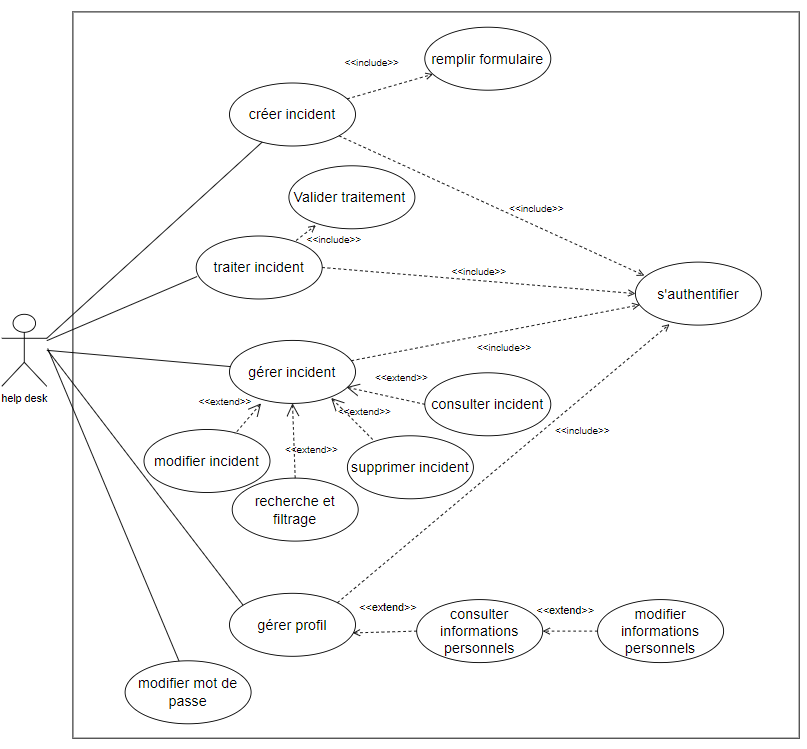


Figure 3 : Diagramme de cas d’utilisation du help-desk

L’operateur peut crée des taches et consulter l’Etat d’avancement en temps réel, peut aussi modifier ou supprimer une taches si besoin. La (figure 7) représente le diagramme de cas d’utilisation de ce rôle.

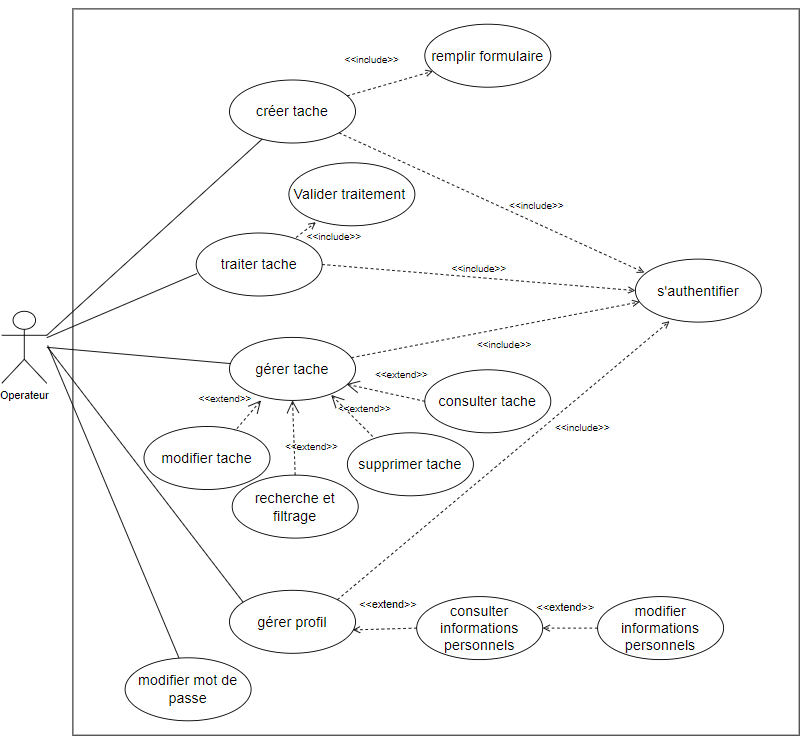


Figure 7 : Diagramme de cas d’utilisation de l’operateur

L’administrateur est en mesure de créer des comptes pour les employées, modifier leurs coordonnées. Il peut également Activer et désactiver les comptes des utilisateurs. La (figure 8) représente le diagramme des cas d’utilisations de ce rôle.

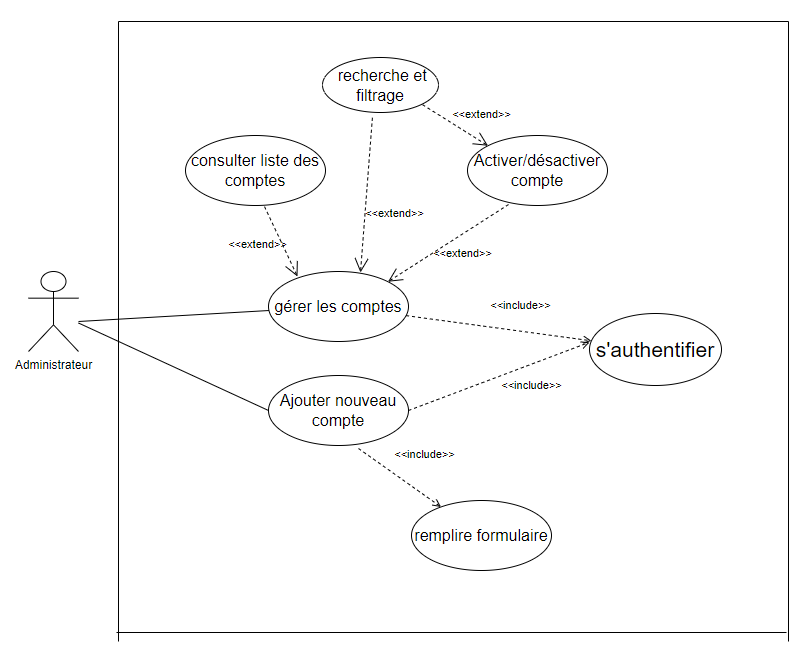


Figure 8 : Diagramme de cas d’utilisation de l'administrateur

L’utilisateur consulte les taches et projet au quelle il participe, traite les taches puis les valide. Il peut également modifier son profil et mot de passe. La (figure 9) représente le diagramme de cas d’utilisation de ce rôle.

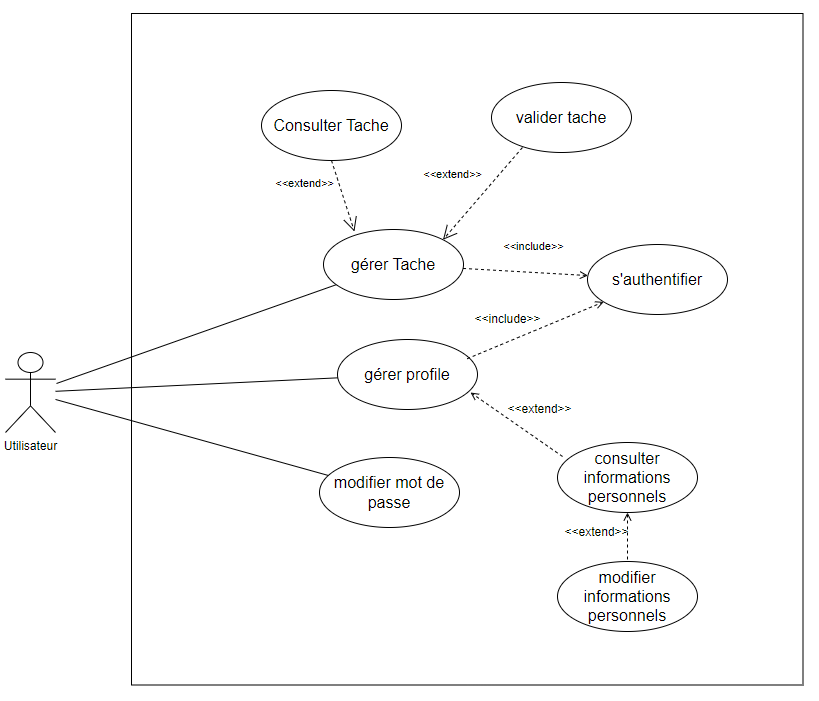


Figure 9 : Diagramme de cas d’utilisation de l'utilisateur

**2.3.3 Diagramme de séquence:**

Le diagramme de séquences est parmi les vues dynamiques d'UML. Il représente les interactions entre les acteurs et le système selon un ordre chronologique

**2.3.3.1 diagramme de séquence authentification :**

Lorsque l’utilisateur souhaite accéder à une session, une page de Login lui est affichée, ou il saisit les coordonnées qui lui ont été attribuées par l’administrateur (nom d’utilisateur, mot de passe). Dès qu’il valide, le système va vérifier l’existence des données introduites, si elles sont valides une session lui est ouverte ou les accès se limitent aux privilèges, sinon ça affiche un message d’erreur. La figure 9 récapitule ce scénario :

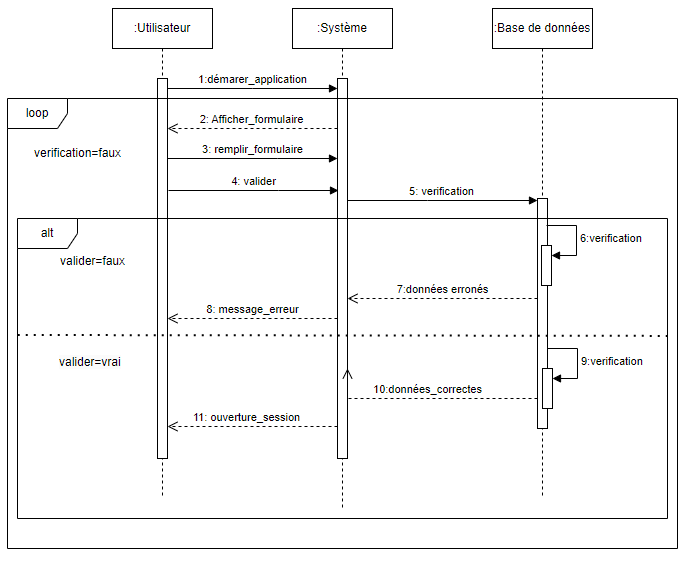


Figure 10 : Diagramme de séquence [Authentification]

**2.3.3.1 diagramme de séquence ajouter projet :**

Pour ajouter un projet, le supérieur doit s’authentifier puis se diriger vers le menu de navigation et choisir « ajouter projet », un formulaire d’ajout s’affiche. Le supérieur remplie le formulaire concernant le projet et clique sur confirmer, un message de succès est affiché en cas d’enregistrement. La figure 10 récapitule ce scénario :

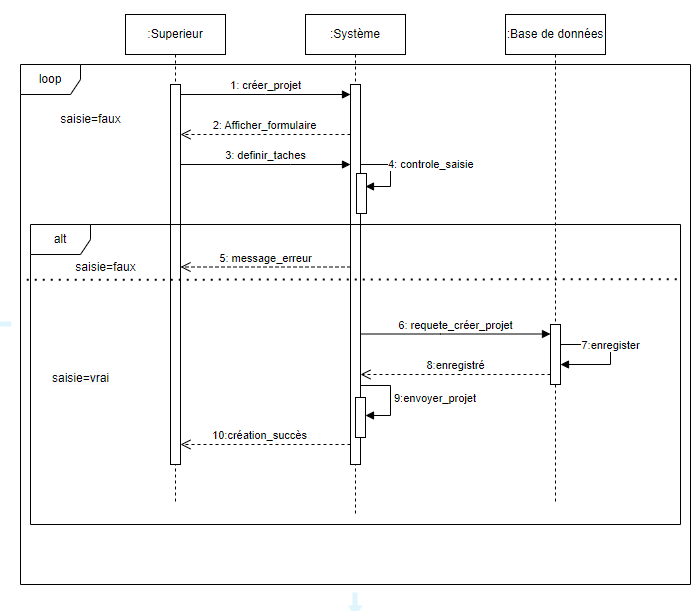
****

Figure 11 : Diagramme de séquence [ajouter projet]

**2.3.3.3 diagramme de séquence ajouter tâche :**

Pour ajouter une tache, l’utilisateur (help-desk ou supérieur) doit s’authentifier puis se diriger vers le menu de navigation et choisir « ajouter tache », un formulaire d’ajout s’affiche. L’utilisateur remplie le formulaire concernant la tâche et clique sur confirmer, un message de succès est affiché en cas d’enregistrement. La figure 11 récapitule ce scénario :

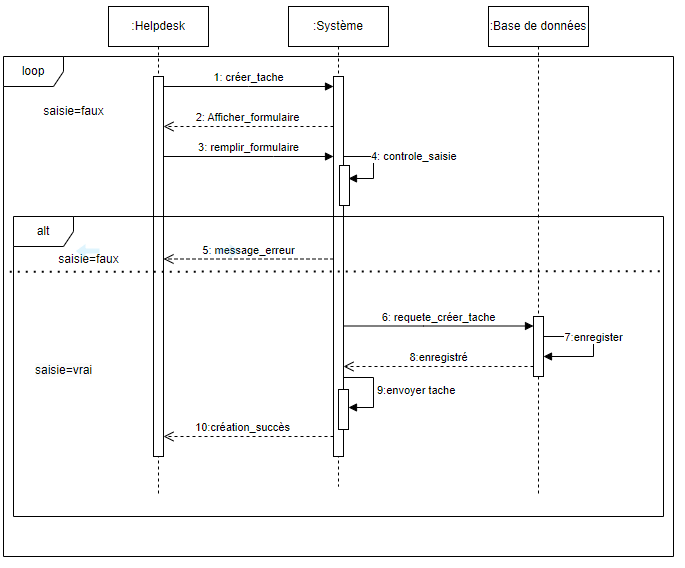
****

Figure 12 : Diagramme de séquence [ajouter tache]

**2.3.3.4 diagramme de traiter tache :**

Pour traiter une tache, l’utilisateur doit s’authentifier puis se diriger vers le menu Dashboard ou il trouvera les taches reçus, après avoir consulter les détails de la tache et réaliser cette dernière l’utilisateur dois la valider en appuyant sur le bouton valider tache. (La figure 12) récapitule ce scénario :

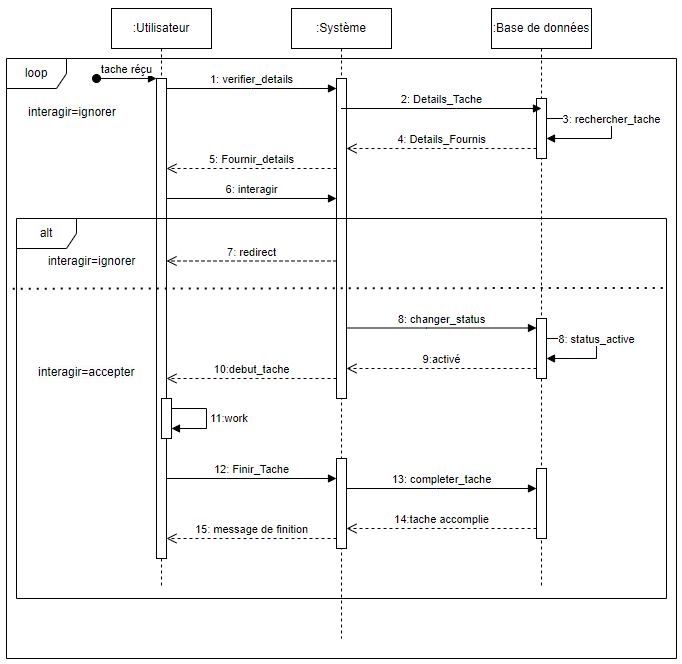
****

Figure 13 : Diagramme de séquence [traiter tache]

**2.3.4 Diagramme de classe :**

Ce diagramme représente la description statique du système en intégrant dans chaque classe la partie dédiée aux données et celle consacrée aux traitements. C’est le diagramme pivot de l’ensemble de la modélisation d’un système. Il permet de fournir une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d’utilisation **.** Comme le montre (la figure 13) :

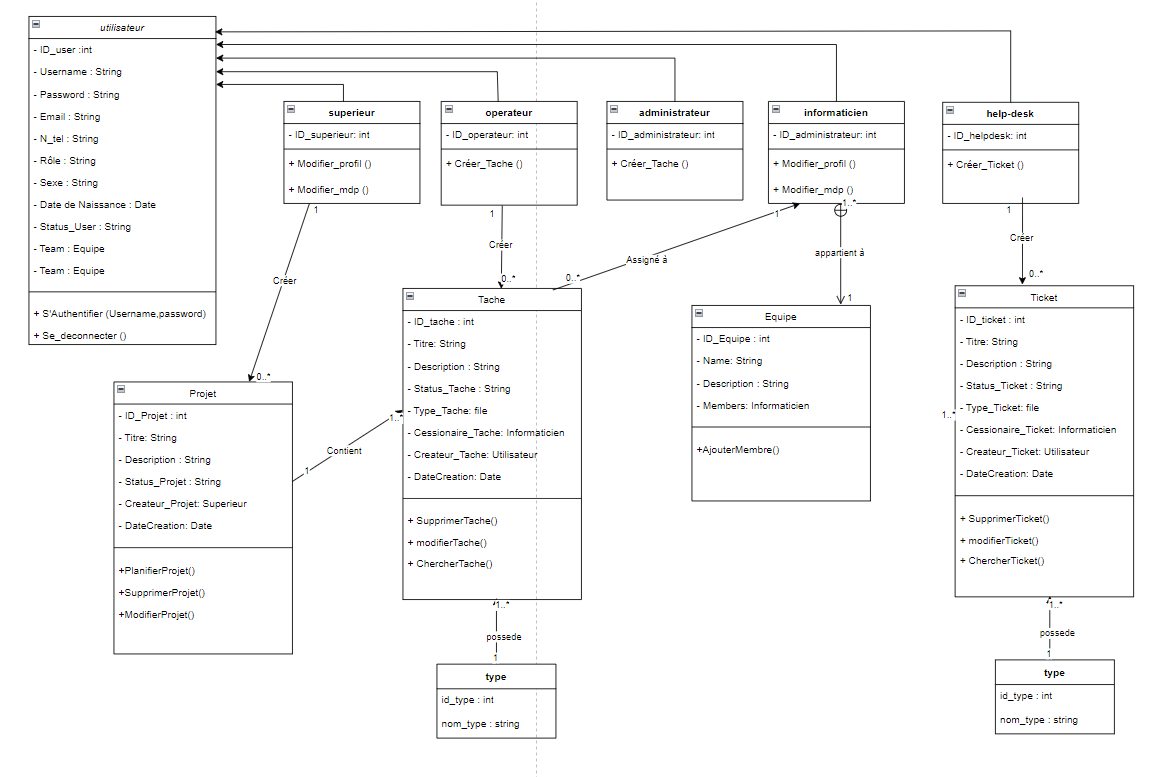


Figure 14: diagramme de classes

**2.3.5 Le schéma relationnel:**

Après l’application des règles de passage de diagramme de classes vers le modèle logique de données, nous avons recensé les relations suivantes :

• Utilisateur (ID\_User\*, Username, Password, Email, N\_Tel, status\_user,Team,sexe

code\_abr\*, ID\_Role\*).

• Superieur (ID\_superieur).

• Operateur (ID\_operateur).

• administrateur (ID\_administrateur).

• Informaticien (ID\_informaticien).

• Help-desk (ID\_help-desk).

• Ticket (ID\_Ticket,Titre\_ticket, Description\_ticket, status\_ticket, ).

• Etat (ID\_type, Nom\_type, ) .