



Rapport Mini-projet

Python-Django

Auteurs :

- Audrey Chauvin
- Gaetan Cren
- Astrid Pierron

I. Sommaire

I. Sommaire.....	1
II. Introduction.....	2
III. Analyse du cahier des charges.....	2
A. Cahier des charges :	2
B. Analyse.....	3
IV. Conception.....	4
A. Base de données.....	4
1. Dictionnaire de données.....	4
Tableau 1 - Dictionnaire de données Projets.....	4
Tableau 2 - Dictionnaire de données Tâches.....	4
Tableau 3 - Dictionnaire de données Projets.....	5
Tableau 4 - Dictionnaire de données Projets.....	5
2. Modèle conceptuel de donnée.....	5
Figure 1 - Modèle Conceptuel des Données.....	5
3. Modèle logique de donnée.....	5
Figure 2 - Modèle logique des données.....	6
B. Diagramme d'identification des acteurs.....	6
Figure 3 - Identification des acteurs.....	6
Figure 4 - Itération du diagramme de cas.....	8
C. Diagramme des cas d'utilisation.....	8
D. Diagramme de classe.....	9
V. Réalisations les plus significatives.....	10
A. Implémentation des classes :	10
VI. Fonctionnalité à venir :	11
VII. Procédure de déploiement.....	11

II. Introduction

Ce document constitue le rapport du développement d'une application destinée à une entreprise ayant besoin d'une application permettant au personnel de suivre un tableau de tâches à réaliser dans le cadre d'un projet. L'entreprise souhaite donc fournir à ses employés une application d'organisation de tâches. Nous l'avons nommé le Système de Gestion de Projets, ou SGP.

III. Analyse du cahier des charges

A. Cahier des charges :

Contexte métier

Une entreprise a besoin d'une application permettant au personnel de suivre un tableau de tâches à réaliser dans le cadre d'un projet. L'objectif est que l'application soit capable de proposer une organisation de tâches la plus efficace en fonction de plusieurs facteurs qui permettront de les prioriser, répartir entre les membres des équipes de développement et les ordonnancer dans le temps.

Un processus de validation des tâches au fur et à mesure de leur statut (planifiée, en cours, réalisée, en pause, validée) permettra aux différent-es responsables d'évaluer l'état d'avancement du projet au regard d'une date de livraison imposée.

Ces responsables de projet auront la possibilité d'avancer ou de reculer la date de livraison et choisissant d'éliminer ou d'ajouter des tâches ou en réduisant ou augmentant le temps de réalisation planifié.

Les responsables sont également chargés de valider une tâche une fois celle-ci réalisée.

Les personnes en charge de tâches doivent faire à rapport quotidien de l'état d'avancement d'une tâche en remplissant un formulaire. Une ou plusieurs personnes peuvent être chargées de la réalisation mais une seule est chargée du rapport sur l'état d'avancement mesuré en pourcentage.

Les personnels, exécutant les tâches ou responsables, peuvent :

- prendre des congés,
- tomber malades,
- arriver en cours de projet,
- partir en cours de projet,
- changer de tâches en cours de réalisation.

Tâches

Une tâche est associée à un projet, à une ou plusieurs personnes exécutantes dont une chargée de l'état d'avancement. Elle dispose d'une description, d'une priorité (sur 3 niveaux), d'une date de démarrage, d'une durée en jours, d'un statut, d'un état d'avancement. Elle peut aussi être associée à une ou plusieurs tâches précédentes nécessaires à sa réalisation.

Toutes les tâches dont les informations nécessaires à sa planification sont manquantes sont par défaut dans l'état « en pause ».

Chaque tâche peut-être également associée à des sous-tâches qui la composent et/ou à une et une seule super tâche dont elle est composée.

Projets

Un projet est associé à plusieurs tâches qui sont nécessaires à sa réalisation ainsi qu'à un-e responsable.

Il dispose d'une date de livraison, qui est fixée par le ou la responsable et d'une date de démarrage, qui correspond à celle de la première tâche planifiée.

Il dispose également d'un statut (en pause, planifié, en cours, livré). Un projet dont l'ensemble des tâches n'ont pas été planifiées est en pause, un projet en cours a été planifié et la date de démarrage de la première tâche planifiée a été passée. Un projet dont l'ensemble des tâches est en pause est également en pause. Un projet est livré si l'ensemble des tâches ont été validées et réalisées.

Il dispose enfin d'un état d'avancement en pourcentage dont le calcul sera laissé à votre appréciation.

Diagramme de GANTT automatisé (Bonus)

L'application sera capable de générer un diagramme de GANTT automatiquement une fois l'ensemble des tâches d'un projet renseignées.

B. Analyse

L'entreprise nécessite une application pour gérer et suivre les tâches d'un projet sous forme de tableau. L'objectif principal est d'optimiser l'organisation des tâches en fonction de divers facteurs. L'application requiert donc les fonctionnalités suivante :

- Suivi et validation des tâches : permettant de suivre les tâches tout au long de leur cycle de vie, de la planification à la validation.
- Priorisation et répartition des tâches : Les tâches doivent être priorisées en fonction de leur importance (de 1 à 3) et attribuées aux membres de l'équipe.
- Gestion du temps : Les tâches doivent être ordonnancées dans le temps en fonction de leur durée en jour.
- Reporting quotidien : Les personnes en charge des tâches doivent fournir un rapport quotidien sur l'état d'avancement en remplissant un formulaire.

Les projets peuvent être en pause, planifiés, en cours ou livrés, en fonction de l'état des tâches qui les composent (planifiée, en cours, réalisée, en pause, validée). L'avancement des projets est mesuré en pourcentage, en fonction du nombre de tâches réalisées et validées par rapport au total des tâches.

Les tâches peuvent être des sous-tâches, associées à une super-tâche parente.

Les mouvements de personnel doivent aussi être pris en compte dans la planification et le suivi des tâches. En effet les employées peuvent :

- Avoir des motifs d'absence :

- prendre des congés
- tomber malades
- Modifier la charge de travail en :
 - arrivant en cours de projet
 - partant en cours de projet
 - changeant de tâches en cours de réalisation.

IV. Conception

A. Base de données

1. Dictionnaire de données

Afin d'établir quelles seront les données nécessaires, nous avons premièrement réalisé un dictionnaire de données sous forme de différents tableaux. Ces tableaux nous serviront à la conception d'une base de données relationnelle.

PROJETS				
Nom de la donnée	Format	Longueur	Type	Remarque
Id	INT		Clé primaire	Identifiant unique
Nom	VARCHAR	255	Chaine de caractère	
Etat d'avancement	FLOAT			%

Tableau 1 - Dictionnaire de données Projets

TÂCHES				
Nom de la donnée	Format	Longueur	Type	Remarque
Id	INT		Clef primaire	Identifiant unique
Nom	VARCHAR	255		
Description	TEXT		Texte	
Durée	TIME			En jour
Etat d'avancement	FLOAT			%
Priorité	INT			

Tableau 2 - Dictionnaire de données Tâches

Utilisateurs				
Nom de la donnée	Format	Longueur	Type	Remarque
Id	INT		Clé primaire	Identifiant unique
Nom	VARCHAR	255		
Role	BOOL			

Tableau 3 - Dictionnaire de données Projets

Dates				
Nom de la donnée	Format	Longueur	Type	Remarque
Id	INT		Clé primaire	Identifiant unique
Début	DATE			
Fin	DATE			
Type	VARCHAR			

Tableau 4 - Dictionnaire de données Projets

2. Modèle conceptuel de donnée

A partir de ces tableaux, nous avons pu définir notre MCD sur looping.

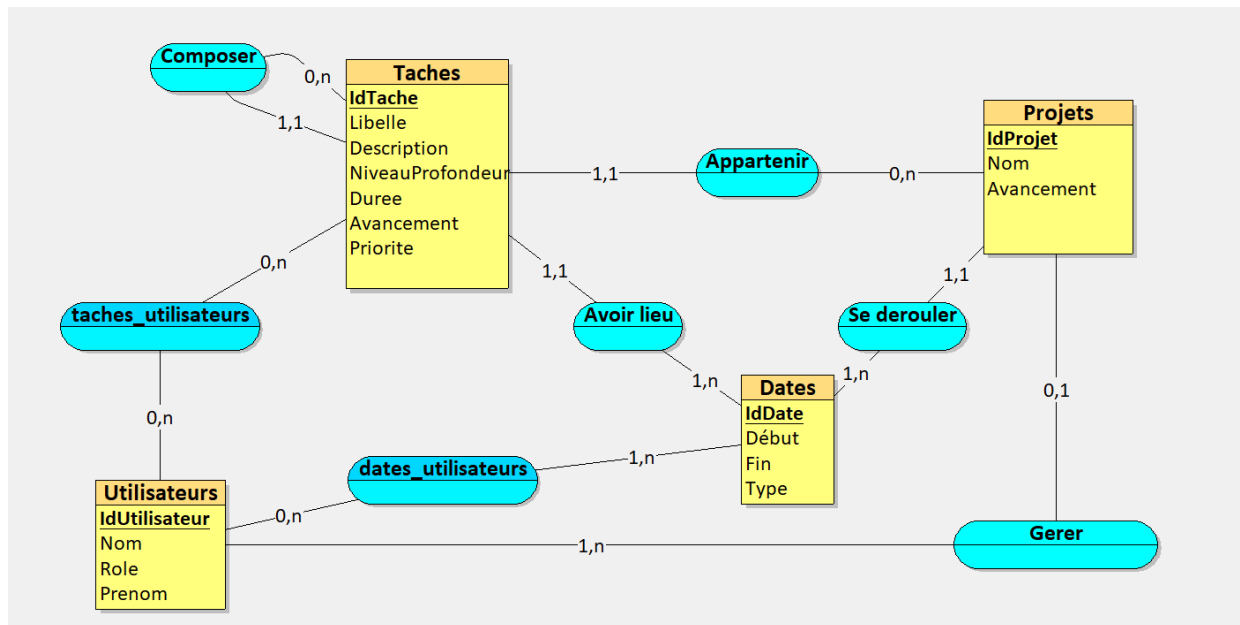


Figure 1 - Modèle Conceptuel des Données

3. Modèle logique de donnée

Du MCD nous avons généré le MLD.

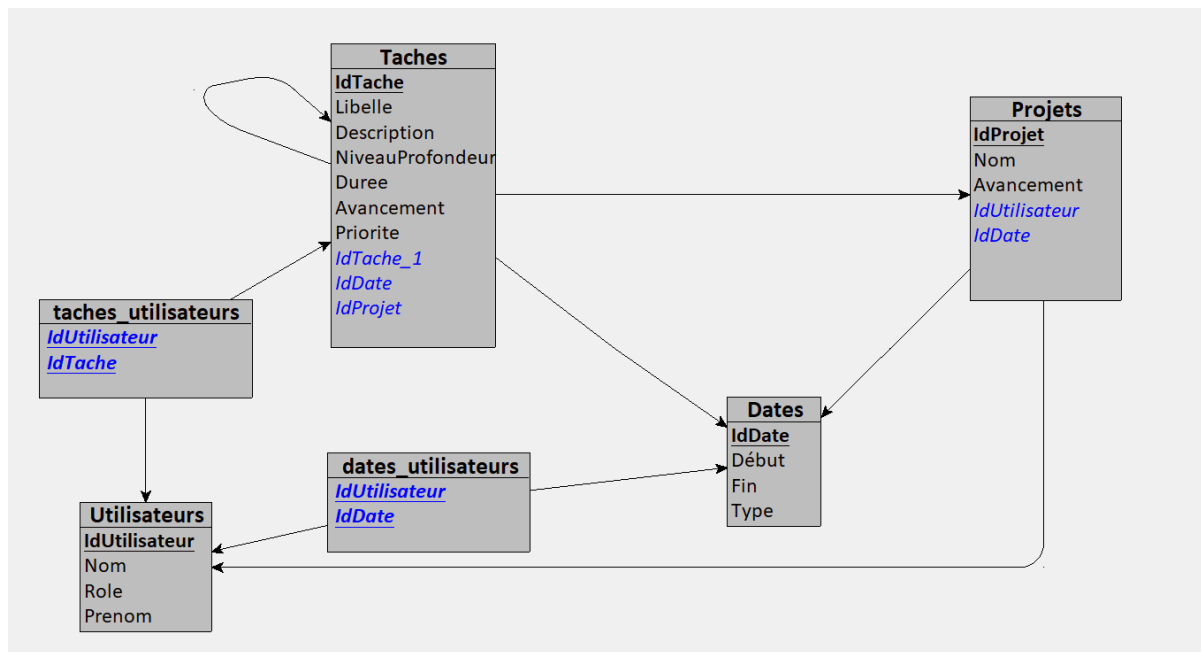


Figure 2 - Modèle logique des données

B. Diagramme d'identification des acteurs

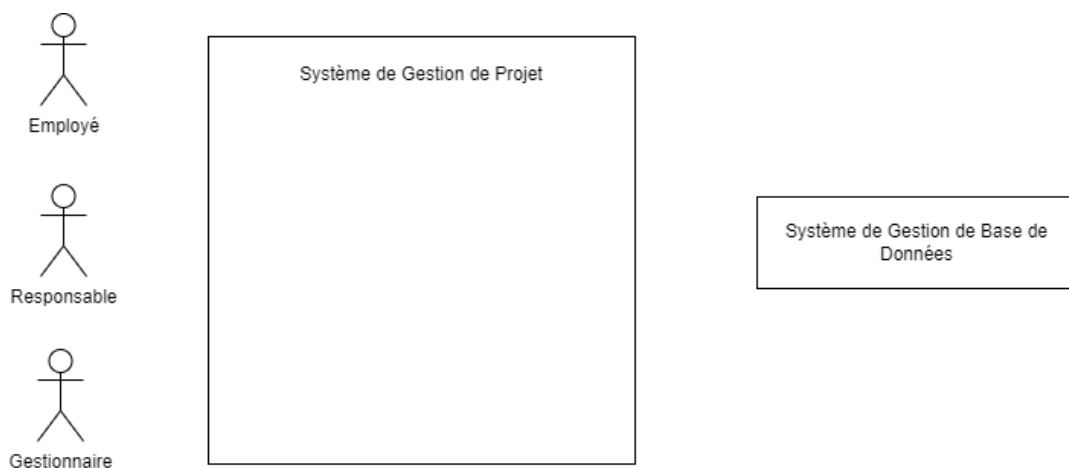


Figure 3 - Identification des acteurs

D'après le cahier des charges, nous pouvons distinguer un acteur principal et deux acteurs secondaires, ainsi qu'un acteur-système et un système.

Acteurs :

- **Employé** : Il s'agit des utilisateurs principaux du système. Un employé peut être aussi :
 - **Responsable** : Il gère le projet et fixe une date de livraison. Il peut s'attribuer des tâches et en attribuer aux employés. Le responsable est également chargé de valider les tâches une fois achevées.
 - **Gestionnaire** : C'est une personne ayant un rôle de référent technique ou PO, chargé du reporting de chaque tâche. Même si plusieurs personnes

peuvent être chargées de la réalisation d'une tâche, un seul gestionnaire est responsable du rapport d'avancement.

Acteur-système :

- **Système de gestion de base de données :** Ce système stocke toutes les données relatives aux projets et aux tâches.
- **Système de gestion des utilisateurs:** Ce système gère à la fois la gestion des utilisateurs et à la fois leur authentification permettant l'accès à l'application. Ce système fait partie des fonctionnalités de django.

Système :

- **Système de configuration de Gestion de Projet :** Le système central de notre projet, permettant aux employés de l'entreprise de gérer les projets et leurs tâches.

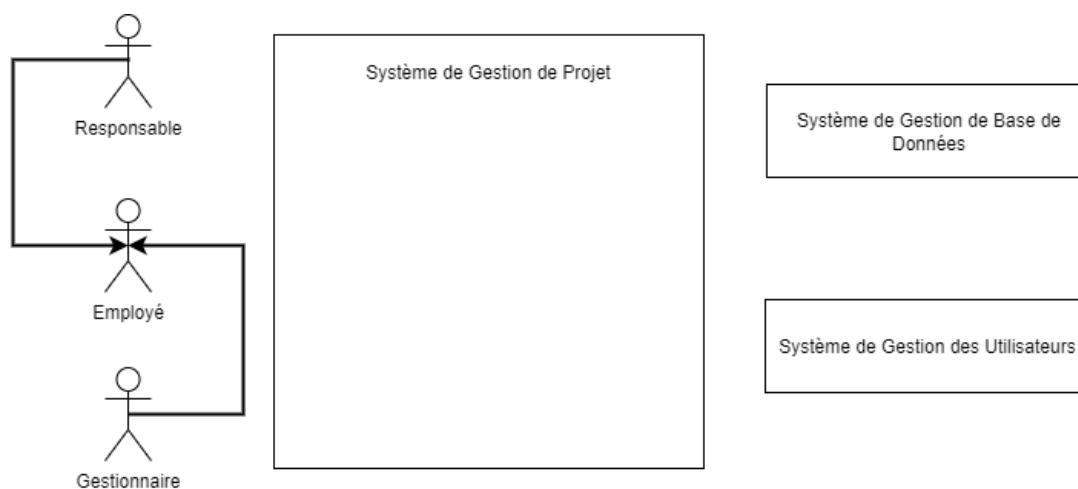


Figure 4 - Itération du diagramme de cas

C. Diagramme des cas d'utilisation

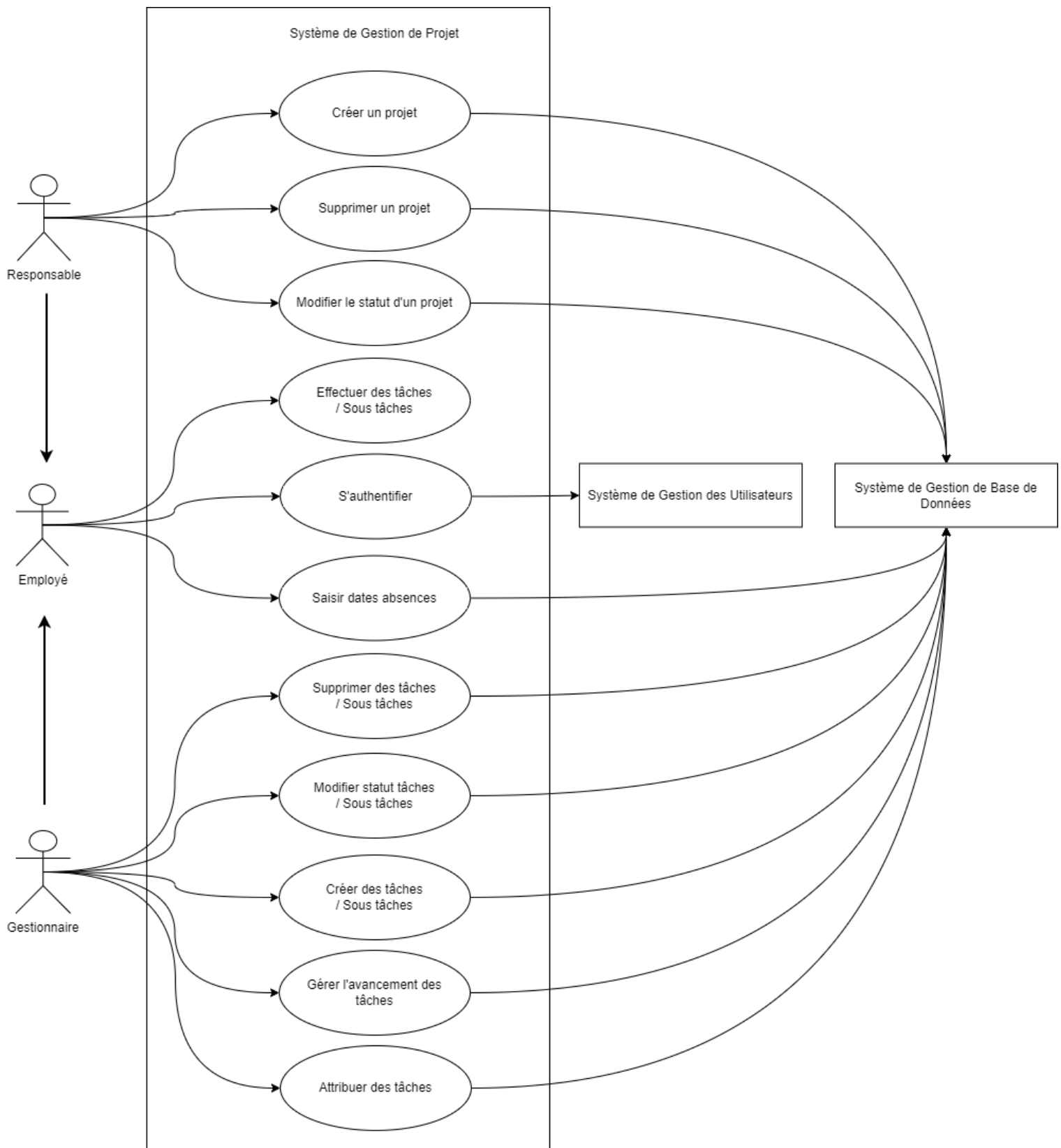


Figure 5 - Diagramme des cas d'utilisation

Notre système comprend plusieurs cas d'utilisations en fonction du User :

- Le responsable : Celui ci peut créer un projet, supprimer un projet et modifier le statut d'un projet. Il peut également effectuer les mêmes actions qu'un employé puisque le responsable reste avant tout un employé (notion d'héritage).
- L'employé : Celui-ci s'authentifie pour avoir l'accès à l'application, il peut effectuer des tâches ainsi que des sous tâches et également saisir ses dates d'absences (congé/arrêt maladie etc..).
- Le gestionnaire : Cet utilisateur peut créer des tâches/sous tâches, il gère l'avancement des tâches, il peut les supprimer, les modifier et également les attribuer à d'autres employés. Il peut aussi effectuer les mêmes actions qu'un employé (notion d'héritage)

D. Diagramme de classe

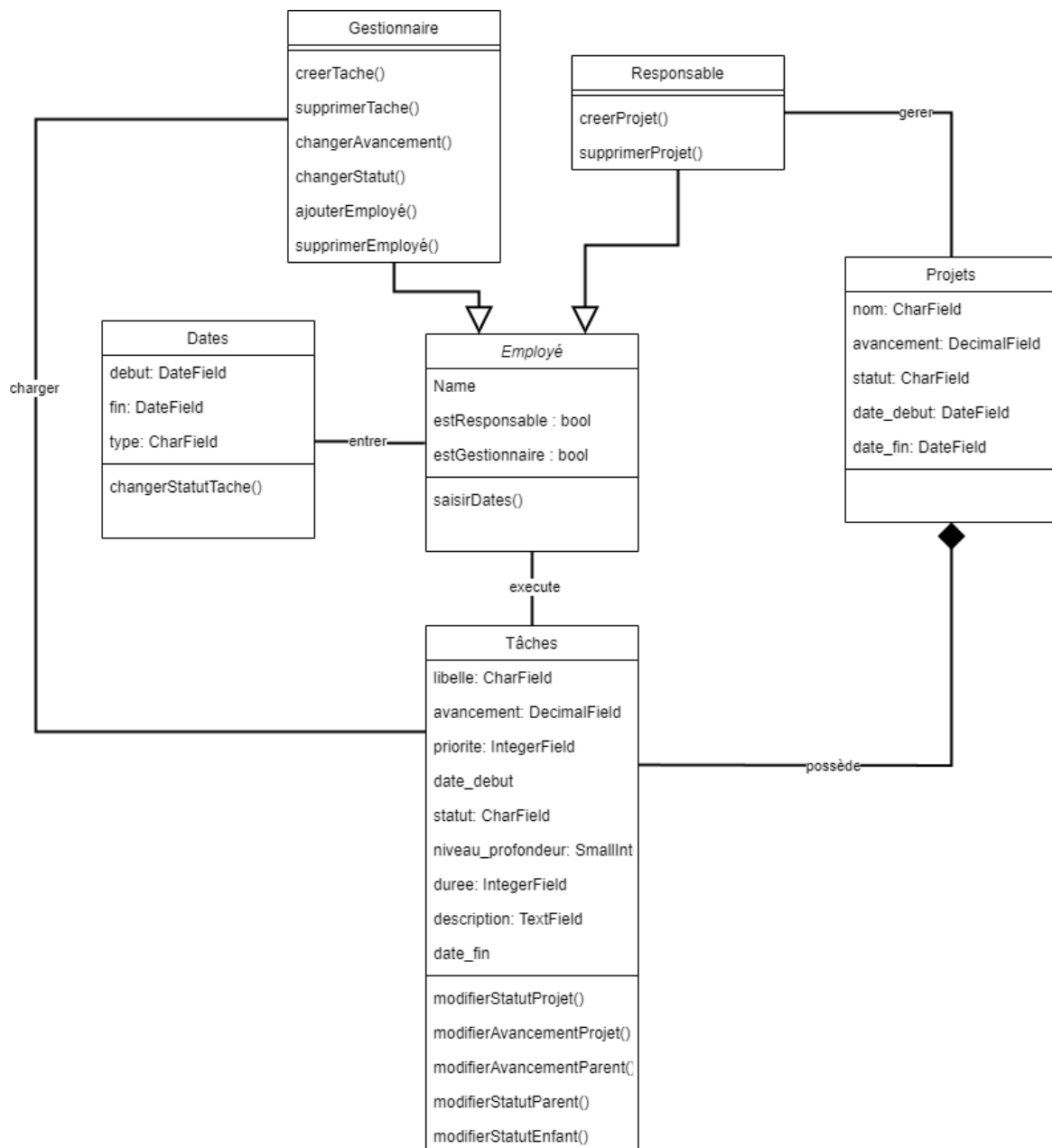


Figure 6 - Diagramme de classe

Au début, nous avons décidé d'utiliser la table 'dates' pour la gestion des dates des tâches et des projets. Cependant, lors de la sauvegarde, Django signale une erreur concernant la perte de mémoire. En réalité, Django empêche, par exemple lors de la sauvegarde d'un projet, la sauvegarde elle-même du projet. La date sera bien présente, mais elle ne sera liée à aucun projet car il n'aura pas pu être créé. Afin de détourner le problème, nous avons décidé de directement inclure les dates de début et de fin dans les tâches et les projets.

Le diagramme de classe ci-dessus représente plus exactement la relation entre les différents éléments du projet.

Nous avons symbolisé, via une composition, la relation entre le projet et les tâches. Si le projet est supprimé, toutes les tâches de ce projet le seront également.

V. Réalisations les plus significatives

- Création d'une application en utilisant une architecture MVC (modèle-vue-contrôleur). Le modèle permet de définir la structure de la base de données. La vue gère la logique de présentation et les interactions de l'utilisateur. La partie contrôleur est prise en compte dans le framework.
- L'application permet la créations de projets ainsi que des tâches.
- Les utilisateurs peuvent également poser des absences.
- Utilisation de la gestion des utilisateurs de django.
- Script qui permet de générer des données.

VI. Fonctionnalité à venir :

Pour une version 2 de l'application nous prévoyons d'implanter d'autres fonctionnalités :

- Le diagramme de gantt (Réalisation par Benjamin)

VII. Procédure de déploiement

Avant toute chose vous devez avoir le protocole de communication sécurisé SSH d'installer sur votre ordinateur. Si ce n'est pas le cas, vous devez l'installer suivant votre distribution. Il vous faudra aussi avoir installé Python sur votre machine virtuelle.

Afin de pouvoir déployer notre application de gestion de projet, il convient de télécharger les fichiers correspondants sous forme de d'archive .zip sur GitHub à l'adresse suivante (commit eaa057a) : [lien](#)

Une fois l'archive téléchargée, dézippez-la à l'endroit de votre choix sur le disque dur. Notez bien cet emplacement, il vous servira à nouveau plus tard.

Démarrez la machine virtuelle.

Une fois celle-ci en route (vous demandant un mot de passe), nous allons maintenant copier le dossier de votre machine hôte vers la machine virtuelle.

Pour ce faire sur votre machine hôte, saisissez cette commande en remplaçant “monchemin” par le chemin où vous avez précédemment extrait l’archive :

Unset

```
scp -r monchemin/evaldjango-master osboxes@192.168.56.12:~/
```

Saisissez le mot de passe de la machine virtuelle : *osboxes.org* si non modifié.

Vous devriez voir une liste de tous les fichiers qui ont été copiés, comme ceci.

```
vcs.xml 100% 180 90.2KB/s 00:0
asgi.py 100% 397 139.1KB/s 00:0
settings.py 100% 3416 352.1KB/s 00:0
urls.py 100% 1027 342.6KB/s 00:0
wsgi.py 100% 397 112.4KB/s 00:0
__init__.py 100% 0 0.0KB/s 00:0
GenererData.py 100% 3402 850.1KB/s 00:0
manage.py 100% 666 166.6KB/s 00:0
admin.py 100% 202 40.6KB/s 00:0
apps.py 100% 164 82.0KB/s 00:0
__init__.py 100% 0 0.0KB/s 00:0
models.py 100% 5831 1.4MB/s 00:0
style.css 100% 2458 618.5KB/s 00:0
style_detail_projet.css 100% 989 247.8KB/s 00:0
style_formulaire_tache.css 100% 375 98.3KB/s 00:0
style_login.css 100% 910 372.9KB/s 00:0
style_projets.css 100% 1040 358.6KB/s 00:0
base.html 100% 770 227.5KB/s 00:0
create_tache.html 100% 1919 353.1KB/s 00:0
detail_projet.html 100% 8050 2.4MB/s 00:0
header.html 100% 846 333.7KB/s 00:0
home.html 100% 447 99.7KB/s 00:0
liste_projets.html 100% 2698 1.0MB/s 00:0
login.html 100% 1853 369.8KB/s 00:0
saisie_absence.html 100% 1180 511.5KB/s 00:0
tests.py 100% 60 13.1KB/s 00:0
urls.py 100% 1132 348.5KB/s 00:0
views.py 100% 10KB 2.1MB/s 00:0
__init__.py 100% 0 0.0KB/s 00:0
PS C:\Users\acrenz |
```

Connectez vous maintenant sur votre machine virtuelle et saisissez le mot de passe depuis la machine hôte :

Unset

```
ssh osboxes@192.168.56.12
```

Une fois que vous êtes connecté et donc à la racine de la machine virtuelle, tapez la commande “ls” pour voir les fichiers et dossiers présents. Vous devriez obtenir ceci :

```
osboxes@osboxes:~$ ls
evaldjango-master
osboxes@osboxes:~$ |
```

Rendez-vous dans le dossier evaldjango-master en saisissant :

```
Unset  
cd evaldjango-master/
```

Une fois dans le dossier, saisissez :

```
Unset  
python3 manage.py makemigrations projetmanagement
```

Ensuite :

```
Unset  
python3 manage.py migrate
```

Et enfin :

```
Unset  
python3 GenererData.py
```

Si tout s'est bien déroulé, votre fenêtre de terminal devrait ressembler à ceci :

```
osboxes@osboxes:~/evaldjango-master$ python3 manage.py makemigrations projetmanagement  
Migrations for 'projetmanagement':  
  projetmanagement/migrations/0001_initial.py  
    - Create model Utilisateur  
    - Create model Projets  
    - Create model Taches  
    - Create model Dates  
osboxes@osboxes:~/evaldjango-master$ python3 manage.py migrate  
Operations to perform:  
  Apply all migrations: admin, auth, contenttypes, projetmanagement, sessions  
Running migrations:  
  Applying contenttypes.0001_initial... OK  
  Applying contenttypes.0002_remove_content_type_name... OK  
  Applying auth.0001_initial... OK  
  Applying auth.0002_alter_permission_name_max_length... OK  
  Applying auth.0003_alter_user_email_max_length... OK  
  Applying auth.0004_alter_user_username_opts... OK  
  Applying auth.0005_alter_user_last_login_null... OK  
  Applying auth.0006_require_contenttypes_0002... OK  
  Applying auth.0007_alter_validators_add_error_messages... OK  
  Applying auth.0008_alter_user_username_max_length... OK  
  Applying auth.0009_alter_user_last_name_max_length... OK  
  Applying auth.0010_alter_group_name_max_length... OK  
  Applying auth.0011_update_proxy_permissions... OK  
  Applying auth.0012_alter_user_first_name_max_length... OK  
  Applying projetmanagement.0001_initial... OK  
  Applying admin.0001_initial... OK  
  Applying admin.0002_logentry_remove_auto_add... OK  
  Applying admin.0003_logentry_add_action_flag_choices... OK  
  Applying sessions.0001_initial... OK  
osboxes@osboxes:~/evaldjango-master$ python3 GenererData.py  
Sample data generated successfully!  
osboxes@osboxes:~/evaldjango-master$ |
```

Votre base de données est maintenant créée et vous avez quelques données qui ont été insérées en guise d'exemple pour vous montrer ce qu'il est possible de faire avec cette application Django de gestion de projets.

3 utilisateurs avec chacun un rôle différent sont enregistrés :

- Astrid avec le rôle de responsable.
- Audrey avec le rôle de gestionnaire.
- Gaëtan avec le rôle d'employé.

Pour ces 3 utilisateurs, le mot de passe est le même, à savoir : *password123*

Vous voudrez peut-être activer le compte administrateur. Pour ce faire, référez-vous à [ce lien](#).

Une fois que vous êtes prêt, nous pouvons démarrer le serveur en saisissant la commande :

```
Unset  
python3 manage.py runserver 0.0.0.0:8000
```

Enfin, rendez vous à l'adresse suivante et appréciez pleinement les fonctionnalités de l'application de gestion de projet : <http://192.168.56.12:8000>

N.B. : Si vous avez créé un compte administrateur vous pouvez accéder à la page admin de Django à l'adresse suivante : <http://192.168.56.12:8000/admin>