

Rapport sur le projet Python

Création d'un logiciel de gestion de médiathèque

Connection administrateur de l'application:

Nom : admintest

Mot de passe : admintest

Connection bibliothécaire à l'application:

Nom : utilisateur1

Mot de passe : mdputi1.

```

class livre():
    name = ""
    auteur = ""
    dateEmprunt = ""
    disponible = ""
    emprunteur = ""

class dvd():
    name = ""
    realisateur = ""
    dateEmprunt = ""
    disponible = ""
    emprunteur = ""

class cd():
    name = ""
    artiste = ""
    dateEmprunt = ""
    disponible = ""
    emprunteur = ""

```

Pour l'édition des class de mon projet, j'ai créé une class Media qui regroupe les champs en commun (name, date_emprunt, le nombre disponible et l'emprunteur) et j'ai transformé les class Livre, Dvd et Cd en class enfant de Media, pour y intégrer les spécificité de chaque class, et en leur rajoutant une majuscule pour respecter la norme du langage python.

J'ai aussi ajouter la fonction `__str__` à chacune des class pour avoir un retour plus lisible et une meilleur maintenabilité dans mon projet, pour l'administration et le débogage.

```

7 usages
class Media(models.Model):
    name = models.CharField(max_length=255)
    date_emprunt = models.DateTimeField(auto_now_add=True)
    disponible = models.IntegerField(default=1)
    emprunteur = models.ForeignKey(Emprunteur, on_delete=models.CASCADE, related_name='emprunts', null=True, blank=True)

    def __str__(self):
        if self.emprunteur:
            return f"Emprunté par {self.emprunteur.name} le {self.date_emprunt.strftime('%Y-%m-%d %H:%M:%S')}"
        else:
            return "Disponible"

12 usages
class Livre(Media):
    auteur = models.CharField(max_length=255)

    def __str__(self):
        return f"{self.name} par {self.auteur}"

12 usages
class Dvd(Media):
    realisateur = models.CharField(max_length=255, null=True, blank=True)

    def __str__(self):
        if self.realisateur:
            return f"{self.name} Réalisé par: {self.realisateur}"
        return f"{self.name}"

12 usages
class Cd(Media):
    artiste = models.CharField(max_length=255, null=True, blank=True)

    def __str__(self):
        return f"{self.name} Artiste: {self.artiste}"

```

```
class jeuDePlateau :
    name = ""
    createur = ""
```

Pour la class JeuDePlateau, j'ai choisi de la laisser sous la même forme, juste de rajouter le type de champ pour pouvoir y stocker des chaînes de caractères pour l'intégration dans la base de données et un paramètre à chacun pour la longueur maximum, que j'ai choisi de laisser au maximum ne sachant pas exactement les longueurs de noms qui pourrait y être intégrer dans le futur.

Et comme pour les class précédentes, j'ai remis au norme du langage en ajoutant la majuscule en début du nom de class.

```
3 usages
class JeuDePlateau(models.Model):
    name = models.CharField(max_length=255)
    createur = models.CharField(max_length=255)
```

```
class Emprunteur():
    name = ""
    bloque = ""
```

Pour la class Emprunteur, j'ai ajouter les champs email, pour l'éventuelle ajout d'une fonction plus tard qui enverrai un mail pour les détails de l'emprunt et la date limite de restitution, et nombre_d'emprunts, car j'ai choisi de gérer le blocage des 3 emprunts au niveau d'un contrôle dans ma fonction « ajouter_emprunt » qui contrôle si le nombre d'emprunts est inférieur ou égal à 3 dans ce champs de la class Emprunteur. Ma fonction « ajouter_emprunt » ajoute 1 au nombre d'emprunts lorsque j'en créer un pour l'emprunteur, et enlève 1 lorsque je déclare le retour du média.

```
16 usages
class Emprunteur(models.Model):
    name = models.CharField(max_length=50)
    email = models.EmailField(unique=True)
    nombre_emprunts = models.IntegerField(default=0)

    def __str__(self):
        return self.name
```

```
def menuBibliotheque() :  
    print("c'est le menu de l'application des bibliothécaire")  
  
def menuMembre():  
    print("c'est le menu de l'application des membres")  
    print("affiche tout")
```

Je n'ai pas utilisé ces fonctions, car telle quelle, elles ne permettent que l'affichage de texte dans le terminal, donc j'ai choisi de faire mes fonctions pour m'approprier le projet avec mes formulations de formule.