Rapport

Dans ce TP nous avons put découvrir JPA et nous avons appris à l'utiliser. Voici le diagramme de classe final de l'application (qui contient les contraintes de la question 5) :

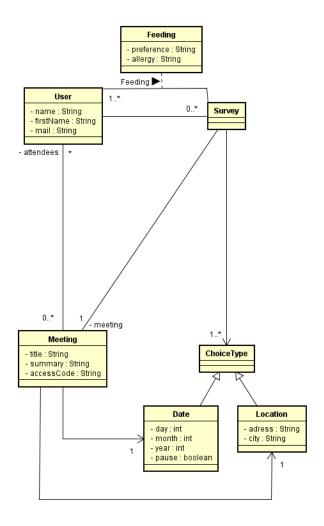


Figure 1: Diagramme de classe TP02

Nous avons ainsi appris à configurer une application JPA, à utiliser le système d'annotations mais nous avons aussi pu mettre en évidence les différences de performance et ainsi voir que le chargement paresseux n'est pas toujours le meilleur (notamment lorsqu'on accède aux attributs liés d'une liste d'objet). On peut donc en conclure qu'il est important de toujours se poser la question de l'utilisation qui sera faite des objets que l'on récupère afin de définir le type de chargement des objets depuis la BDD. Voici les captures d'écran de la différence de performance :

```
Hibernate: select employees0.department id as department3_1, employees0.id as id0_1, employees0.id as id0_0, employees0.department id as department3_0_0, employees0.name as name0_0 from Hibernate: select employees0_department id as department3_1_1, employees0_id as id0_1, employees0_id as id0_0, employees0_department id as department3_0_0, employees0_name as name0_0 from Hibernate: select employees0_department id as department3_1_1, employees0_id as id0_1, employees0_id as id0_0, employees0_department id as department3_0_0, employees0_name as name0_0 from Hibernate: select employees0_department id as department3_1_1, employees0_id as id0_1, employees0_id as id0_0, employees0_department1_id as department3_0_0, employees0_name as name0_0 from Hibernate: select employees0_department3_0_0, employees0_name as name0_0 from ... done
temps_d'exec = 2185

Process finished with exit code 0
```

Figure 2: Chargement N+1 Select (temps: 2189ms)

```
INFO: HIML000109: Everign keys: []
mars 07, 2019 2:5912 EM (eys.) []
mars
```

Figure 3:Chargement JoinFetch (temps: 438ms)