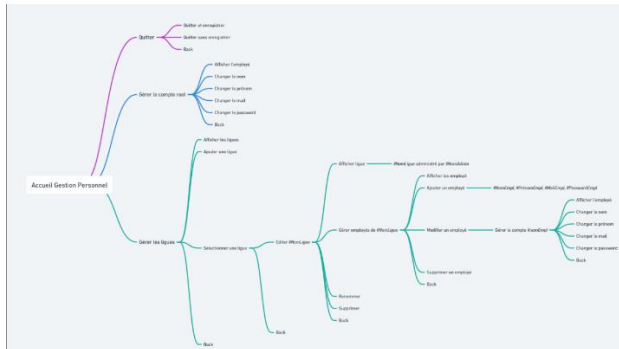
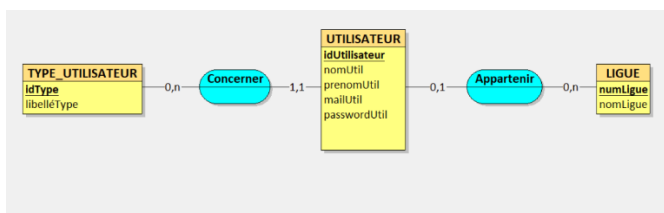


Documentation itération1

Représentation de l'application de gestion d'employé d'une ligue avec un arbre heuristique.



MCD représentant les données de l'application.



Ce code définit une exception personnalisée DateIncoherente, qui est utilisée pour signaler une erreur liée à une date incohérente. Il écrit donc un message d'erreur.

```

    */
    private static final long serialVersionUID = 1L;

    - public DateIncoherente(String message) {
    -     super(message);
    + public String DateIncoherente() {
    +     return "";
    }

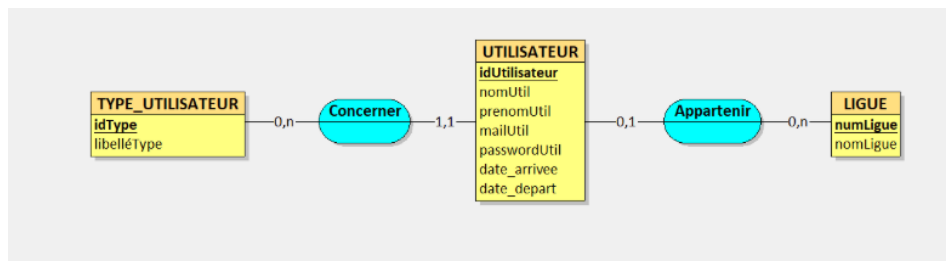
}

```

Test unitaire vérifiant le fonctionnement des dates.

```
@Test // test pour vérifier les dates incohérentes
void testDate(){
    Ligue ligue = gestionPersonnel.addLigue("Fléchettes");
    assertThrows(DateIncoherente.class, () -> {
        ligue.addEmploye("Tojo", "Tojo", "tojo@gmail.com", "password046", LocalDate.of(2024, 12, 01), LocalDate.of(2022, 12, 01));
    });
}
```

Modification du mcd ajoutant la date de départ et d'arrivée



Ce code est un test unitaire qui vérifie la création d'une ligue dans un système de gestion du personnel.

```
class testLigue {
    GestionPersonnel gestionPersonnel = GestionPersonnel.getGestionPersonnel();

    //
    @Test
    void createligue() throws SauvegardeImpossible {
        Ligue ligue = gestionPersonnel.addLigue("Fléchettes");
    }
}
```