DOCUMENTATION DEVELOPPEUR
Attribution du suivi des étudiants

TRAN DIEP Mai Thi

Table des matières

I. Contexte :

Un des responsables de la M2L, utilise une application pour gérer les employés des ligues. Cette application, très simple, n'existe qu'en ligne de commande et est mono-utilisateur. Nous souhaiterions désigner un administrateur par ligue et lui confier la tâche de recenser les employés de sa ligue.
L'application JAVA a pour but de gérer les employés des ligues. Pour cela trois niveaux d'habilitation ci-dessus doivent être mis en place:

- Un simple employé de ligue peut ouvrir l'application et s'en servir comme un annuaire, mais il ne dispose d'aucun droit d'écriture.
- Un employé par ligue est administrateur et dispose de droits d'écriture peut gérer la liste des employés de sa propre ligue avec une application bureau.
- Le super-administrateur a accès en écriture à tous les employés des ligues. Il peut aussi gérer les comptes des administrateurs des ligues avec une application accessible en ligne de commande.

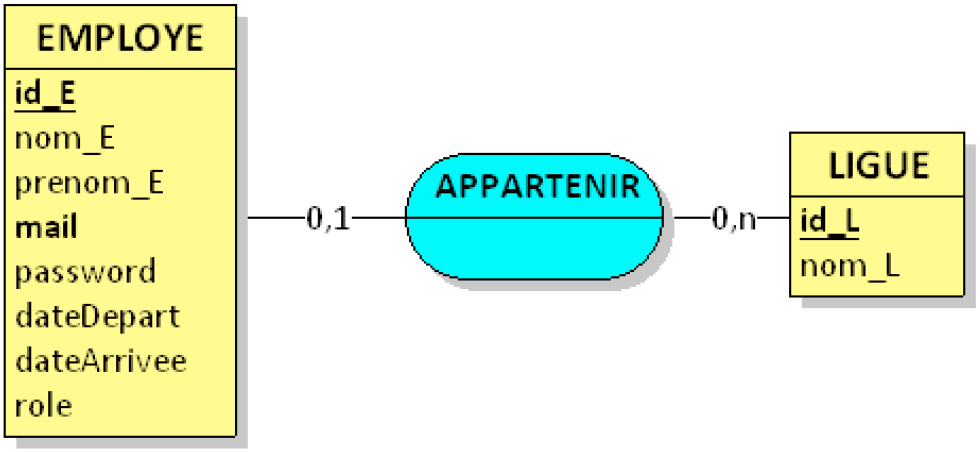
II. Versionning

Afin de sauvegarder toute modification effectuée sur le code, nous avons utilisé l'outil Git, qui permet d'envoyer le projet et de sauvegarder toute modification sur le repository GitHub. Grâce au repository GitHub et git, nous pouvons annuler toute modification et revenir en arrière lorsqu'un bug se produit, ou une fonctionnalité n'est pas voulue.

Il est possible d'y accéder par ce lien : https://github.com/Mel-Red/Personnel

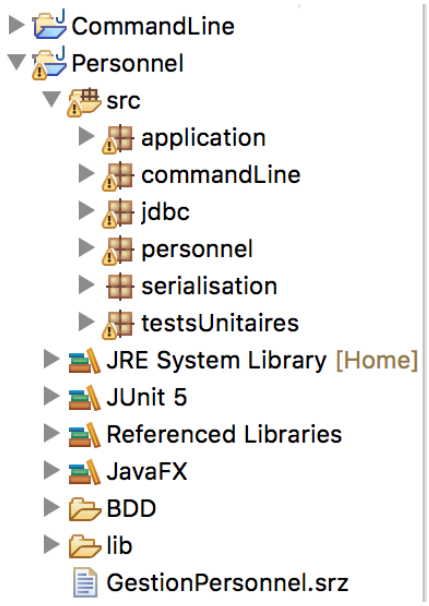
III. Base de données :

MCD



IV. Spécifications fonctionnelles :

V. Composants logiciels :



- Le dossier CommandLine est une bibliothèque logiciel de dialogue en ligne de commande.
- Le dossier jdbc est le dossier l'un des plus important, c'est dans celui-ci qu'on stocke la connexion à la base de données, des fonctions, traitement des requêtes.
- Le dossier serialization contient une classe Serialization qui permet à convertir un objet en un flux d'octets pour stocker l'objet ou le transmettre à la mémoire, à une base de données ou à un fichier.
- Le dossier testsUnitaires contient une classe Test pour tester les fonctionnalités de l'application.
- La classe CredentialsExample se trouvant dans le dossier jdbc est à éditer lors de la mise en place du projet sur une autre base de données, il faut alors modifier les varaibles : host, port, database, user, password avec vos informations de connexions.

VI. Testes unitaires :

1. Test si un employé est administrateur

```
@Test
void testEstAdmin() throws SauvegardeImpossible, ImpossibleDeChangerDate
{
    Ligue ligue = gestionPersonnel.addLigue("L1");
    Employe employe = ligue.addEmploye("ToTo", "HoHo", "toto.hoho@gmail.com", "hello", null, null);
    assertEquals(employe.estAdmin(ligue), false);
    ligue.setAdministrateur(employe);
    assertEquals(employe.estAdmin(ligue), true);
    Ligue ligue2 = gestionPersonnel.addLigue("L2");
    Employe employe2 = ligue2.addEmploye("DoDo", "HoHo", "DoDo.hoho@gmail.com", "hello", null, null)
    assertEquals(employe2.estAdmin(ligue), false);
}
```

2. Test de créer/supprimer un employé

```
@Test
void addEmploye() throws SauvegardeImpossible, ImpossibleDeChangerDate
{
    Ligue ligue = gestionPersonnel.addLigue("Fléchettes");
    Employe employe = ligue.addEmploye("Bouchard", "Gérard", "g.bouchard@gmail.com", "azerty", null,null);
    assertEquals(employe, ligue.getEmployes().first());
}

@Test
void removeEmploye() throws SauvegardeImpossible, ImpossibleDeChangerDate
{
    Ligue ligue = gestionPersonnel.addLigue("L1");
    Employe employe = ligue.addEmploye("ToTo", "HoHo", "toto.hoho@gmail.com", "hello", null, null);
    ligue.remove(employe);
    assertEquals(ligue.hasEmploye(employe), false);
}
```

3. Test de créer/supprimer une ligue

```
@Test
void createLigue() throws SauvegardeImpossible
{
    Ligue ligue = gestionPersonnel.addLigue("Fléchettes");
    assertEquals("Fléchettes", ligue.getNom());
}

@Test
void setLigue() throws SauvegardeImpossible
{
    Ligue ligue = gestionPersonnel.addLigue("L1");
    ligue.setNom("New L1");
    assertEquals(ligue.getNom(), "New L1");
}

@Test
void removeLigue() throws SauvegardeImpossible
{
    Ligue ligue = gestionPersonnel.addLigue("ToBeDeleted");
    gestionPersonnel.remove(ligue);
    assertEquals(gestionPersonnel.hasLigue(ligue), false);
}
```