AP - Personnel

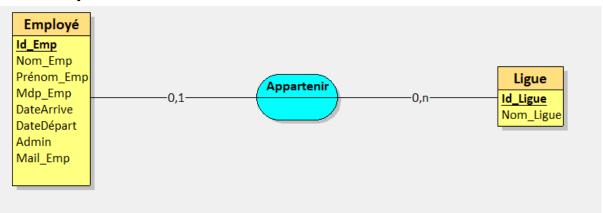
Avec notre groupe d'AP nous avons modifié une application Java réalisé par notre professeur en faisant un Fork via git (git clone)

```
KADEY@DESKTOP-J3AFIU5 MINGW64 ~/Desktop/Kadeyofficiel.github.io (main)
$ git clone + adresse du dépot
```

Les modifications réalisés pour l'instant sont :

- La gestion des ligues (insertion, modification, suppression)
- La gestion des employés (insertion, modification, suppression)
- La gestion de l'administrateur (en ligne de commande/interface du terminal)
 - + La gestion des dates, mot de passes, tests unitaires
- La création la base de données correspondante et du lien entre les deux (NON FINALISÉ)

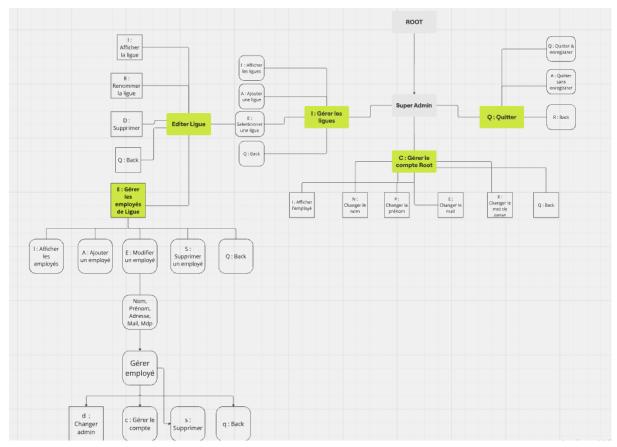
Création de la base de donnée (MCD + SCRIPT SQL) & Arbre heuristique



Lien vers le commit du MCD

```
DROP TABLE if exists Ligue;
DROP TABLE if exists Employe;
CREATE TABLE Ligue(
   ID_LIGUE INT,
    NOM_L VARCHAR(),
    PRIMARY KEY(ID_LIGUE)
)engine = innodb;
CREATE TABLE Employe(
   ID_EMP INT,
    NOM_EMP VARCHAR(),
    PRENOM_EMP VARCHAR(),
    MDP_EMP VARCHAR(),
    DATE_ARRIVE DATE(),
    DATE_DEPART DATE(),
    ADMIN BOOLEAN,
    MAIL_EMP VARCHAR(),
    PRIMARY KEY(ID_EMP),
    FOREIGN KEY(ID_LIGUE) references LIGUE(ID_LIGUE)
)engine = innodb;
```

Lien vers le commit du script SQL



Lien vers le commit de l'arbre heuristique

2 / L'application Personnel

```
🕶 🚜 > src

▼ 

■ > commandLine

     > 13 > EmployeConsole.java
    > 🋂 > LigueConsole.java
     > 13 PersonnelConsole.java
  ▼ 👼 > jdbc
     > 🗾 Credentials.java
     > 4 CredentialsExample.java
     > 🛂 > JDBC.java

▼ 35 > personnel

     > 13 DroitsInsuffisants.java
    > 🛂 > Employe.java
     > A ExceptionArrivee.java
     ExceptionDepart.java
     > 🎎 > GestionPersonnel.java
     > 4 ImpossibleDeSupprimerRoot.java
     > 🛂 > Ligue.java
     > 14 > Passerelle.java
     > 🛂 SauvegardeImpossible.java

▼ 

■ > serialisation

     > 🛂 > Serialization.java

▼ # > testsUnitaires

     > 1/2 testEmploye.java
     > 14 testGestionPersonnel.java
    > 🋂 > testLigue.java
```

```
Gestion du personnel des ligues
c : Gerer le compte root
```

l : Gerer les ligues

q : Quitter

Select an option :

```
D X
eclipse-workspace - Personnel/src/commandLine/LigueConsole.java - Eclipse IDE
Fichier Éditer Code source Reconcevoir Navigation Rechercher Projet Exécuter Fenêtre Aide
Explorateur de packages X
                                       🗀 🔟 LigueConsole.java 🗴 Employe.java 📝 Ligue.java 📝 Erreurdate.java 📝 PersonnelConsole.java
                                                                                                                                                         - -
                                                         private List<Employe> GererEmploye(final Ligue ligue)
{
                             E & $
                                                 1448

→ Big Personnel [AP-Parking main]

                                                              return new List<Employe>("Gérer un employé".
                                                 146
  () -> new ArrayList<>(ligue.getEmployes()),
(element) -> gerer7employe(element)
                                                 147
     ∨ ∰ commandLine
       > [] EmployeConsole.java
> [] LigueConsole.java
                                                 150
                                                          }
                                                 151
       > A PersonnelConsole.java
                                                          private Menu gerer7employe(final Employe employe)
     > 🚜 jdbc
                                                 153
     > 🔠 personnel
                                                              Menu menu = new Menu("Gérer un employé " + employe.getNom(), "m");
                                                 154
                                                               menu.add(modifierEmploye(employe));
menu.add(supprimerEmploye(employe));
menu.add(changerAdministrateur(employe));
    > ## serialisation
                                                  155
                                                 156
157
     > 📠 testsUnitaires
  > M Bibliothèque système JRE [JavaSE-1.8]
                                                 158
                                                               menu.addBack("q");
  > M JUnit 5
                                                 159
160
  > Maven Dependencies
    target
                                                 161

    ■ GestionPersonnel.srz

                                                  1628
                                                          private Option changerAdministrateur( final Employe employe)
                                                 163
164
    pom.xml
                                                              return new Option("Changer admin" , "d" , () -> adm(employe));
                                                          1
                                                 165
                                                  166
                                                          private void adm(Employe employe) {
   employe.getLigue().setAdministrateur(employe);
   System.out.println(employe.getNom() + " " + employe.getPrenom() + " " + "est maintenant");
}
                                                 168
                                                 169
                                                 170
171
                                                          private Option supprimer(Ligue ligue)
                                                 172⊖
                                                 173
174
175
176
                                                              return new Option("Supprimer", "d", () -> {ligue.remove();});
                                                 1778
                                                          private Option supprimerEmploye(final Employe employe)
                                                              return new Option("Supprimer " + employe.getNom() + " " + employe.getPrenom(), "s",
                                                  188
                                                  181
                                                                           employe.getLigue().getEmployes().remove(employe);
                                                          }
                                                  183
                                                 184
                                                  185
                                                          private Option modifierEmploye( final Employe employe)
                                                 187
                                                 188
                                                              return employeConsole.editerEmploye(employe);
                                                      <
                                                                                                          📘 Problèmes @ Javadoc 😡 Declaration 🖾 Console 🗶
                                               PersonnelConsole [Application Java] C:\Users\mkada\.p2\poof\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32x86_64_23.0.0.v2024091
                                               Gérer les employés de foot
1 : Afficher les employes
a : ajouter un employé
x : Gérer un employé
                                               q : Back
                                               Select an option :
<
                                                     Inscriptible Insertion avancée 90 : 33 : 2193
```

Ensuite nous avons décidé <u>d'améliorer la prise en considération des erreurs</u>, notamment concernant les dates d'arrivés et de départs, par exemple concernant le format de la date.

```
private Option ajouterEmploye(final Ligue ligue)
     return new Option("ajouter un employé", "a",
                () ->
                      // Saisie en chaîne - demande toutes les informations d'abord
String nom = getString("nom : ");
String nonnem = getString("nom : ");
                      String prenom = getString("prenom : ");
String mail = getString("mail : ");
                      String password = getString("password : ");
String dateArrStr = getString("date arrivée (AAAA-MM-JJ) : ");
String dateDepStr = getString("date départ (AAAA-MM-JJ) : ");
                            // Exception Format - <u>vérifie le</u> format <u>des</u> dates
                            LocalDate dateArrivee = LocalDate.parse(dateArrStr);
                            LocalDate dateDepart = LocalDate.parse(dateDepStr);
                            // Exception ordre - vérifie l'ordre chronologique
                            if (dateDepart.isBefore(dateArrivee)) {
                                 throw new Erreurdate();
                            // Si tout est valide, crée l'employé
                           ligue.addEmploye(nom, prenom, mail, password, dateArrivee, dateDepart);
System.out.println("Employé ajouté avec succès");
                      } catch (DateTimeParseException e) {
                            // Gestion de l'exception de format
System.out.println("Erreur : Format de date invalide. Utilisez le format AAAA-MM-JJ");
                      } catch (Erreurdate e) {
    // Gestion de l'exception d'ordre chronologique
                            System.out.println(e.getMessage());
                }
     );
}
```