UtilisateurDAO :

Classes :

Utilisat

* getRole(courriel, mdp) //renvoie le rôle de l’utilisateur qui se connecte, pour différencier les vues

return “Jury” ou “Gest” ou une erreur

* getInfoCompte(courriel)

return un Utilisateur

* updateMdp(oldMdp, newMdp, courriel)

return ok ou erreur

CandidatDAO :

*//envoi de mails sous java si le temps*

* *MailsAdmissibles() //recupere les mails pour l’envoi d’un mail aux admissibles*

*return liste avec (Nom, Prenom, courriel)*

* *MailsAdmis() //recupere les courriels pour l’envoi d’un mail aux admis*

*return liste avec (Nom, Prenom, courriel)*

* *MailsRefusés1() //recupere les courriels pour l’envoi d’un mail aux non admissibles*

*return liste avec (Nom, Prenom, courriel)*

* *MailsRefusés2() //recupere les courriels pour l’envoi d’un mail aux non admis qui étaient admissibles*

*return liste avec (Nom, Prenom, courriel)*

[ Pour Thibault ?

* getInfoDossier() //consulter les dossiers des candidats (cad lettre de motivation cv etc.. (à rediscuter pour quelle est forme les fichiers : telecharger des pdf ?, zip ?, visualisation du pdf dans la vue ?)

return liste avec (Nom, Prenom, Dossier, courriel) ]

* getCandidatsAdmissibles()

return liste (Candidat)

* getCandidatsAdmis()

return liste (Nom Prenom “””Dossier”””)

* getCandidat(id) // rend toutes les infos du candidat “id”
* getAllCandidat() //rend une ArrayList de tous les candidats
* changeCandidatAdmis(candidat)//rend un candidat admis
* changeCandidatAdmissible(candidat)//rend un candidat admissible

EntretienDAO :

* getInfoNote() //permet au jury et au gestionnaire de voir les candidats et rentrer les notes par la suite

return liste(Nom, Prénom, Date, Heure, Salle, Note, courriel)

* setNote(courriel(candidat), note)//mettre une note suite a l’entretien
* getNot(courriel(candidat)) //consulter la note de l’entretien
* setHoraire(idcandidat, idJury, nouvelHoraire) //changer l’horaire d’un entretien
* getInfoEntretien(idJury) //liste des entretien du jury

return liste(Nom, Prénom, Date, Heure, Salle, courriel) entretien ?

* getAllEntretien() // liste de tous les entretiens
* nextMeeting(courriel) //info du prochain entretien

return Entretien avec (salle, date, heure, nom candidat, prénom candidat)

JuryDAO :

* getAllJury() //rend une ArrayList d’id de jury

Salle :

* getAllSalles() //rend une ArrayList de toutes les salles

public void setHoraireDebut(int idcandidat, Date dateDebut) throws SQLException {

String requete = "UPDATE Entretien SET dateDebut=" + dateDebut + " WHERE Candidats\_idCandidat=" + idcandidat;

int rs = this.statement.executeUpdate(requete);

}

public void setHoraireFin(int idcandidat, Date dateFin) throws SQLException {

String requete = "UPDATE Entretien SET dateFin=" + dateFin + " WHERE Candidats\_idCandidat=" + idcandidat;

int rs = this.statement.executeUpdate(requete);

}

public Entretien nextMeeting(int idJury) throws SQLException {

ResultSet rs = this.statement.executeQuery("SELECT Candidats\_idCandidat,nom,prenom,courriel,dateDebut,dateFin,idSalle,Jury\_iddJury FROM Utilisateurs, Candidats, Entretiens, Salles, Jury WHERE Candidats\_idCandidat=fk\_idCandidat AND fk\_idCandidat=idUtilisateur AND fk\_idSalle=idSalle;");

Date minDate = new Date(2100,12,31);

boolean trouve = false;

int idCandidat=0;

Date dateDebut=null;

Date dateFin=null;

int idSalle=0;

while (rs.next()) {

int jury = rs.getInt("Jury\_idJury");

if(idJury == jury){

trouve = true;

dateDebut = rs.getDate("dateDebut");

if(minDate.after(dateDebut)){

minDate = dateDebut;

idCandidat = Integer.parseInt(rs.getString("Candidats\_idCandidat"));

dateFin = rs.getDate("dateFin");

idSalle = Integer.parseInt(rs.getString("idSalle"));

}

}

}

if(trouve) {

Entretien entretien = new Entretien(idCandidat,dateDebut,dateFin);

entretien.setIdSalle(idSalle);

System.out.println(entretien.toString() + "envoyé");

return entretien;

}else {

System.out.println("échec de l'envoi du prochain meeting");

return null;

}

}

//autre proposition setNote

public void setNote(int idcandidat,Date dateDebut, int note) throws SQLException {

String requete = "UPDATE Entretien SET note=" + note + " WHERE Candidats\_idCandidat=" + idcandidat + " AND dateDebut = " + dateDebut;

int rs = this.statement.executeUpdate(requete);

}

//getAllNomSalles()

public List<String> getAllNomSalles() throws SQLException {

ResultSet rs = this.statement.executeQuery("SELECT nomSalle FROM Salles;");

List<String> salles = new ArrayList<String>();

while(rs.next()) {

String nomSalle = rs.getString("nomSalle");

salles.add(nomSalle);

}

return salles;

}

//setEntretien dans Entretien\_daoimplServeur

public void setEntretien(int idCandidat,String nom, String prenom, Date dateDebut, Date dateFin,int idJury,int idSalle) throws SQLException {

String requete = "INSERT INTO Entretien VALUES (" + idCandidat + " ," + idJury + " ," + dateDebut + " ," + dateFin + " ," + idSalle + ");";

int rs = this.statement.executeUpdate(requete);

}

//Classe InfoJury pour getAllJury()

package app.businessObject;

import java.util.Date;

public class InfoJury {

private String nom;

private String prenom;

private int idJury;

public InfoJury(String nom, String prenom, int idJury) {

this.nom = nom;

this.prenom = prenom;

this.idJury = idJury;

}

}

//getAllJuree() dans Jury\_DAOimplServeur

public List<InfoJury> getAllJuree() throws SQLException {

ResultSet rs = this.statement.executeQuery("SELECT nom,prenom,Jury\_idJury FROM Utilisateurs,Fait\_Partie\_De,Jury WHERE Utilisateurs\_idUtilisateur=idUtilisateur;");

List<InfoJury> listeJuree= new ArrayList<InfoJury>();

while (rs.next()) {

int idJury = Integer.parseInt(rs.getString("Jury\_idJury "));

String nom = rs.getString("nom");

String prenom = rs.getString("prenom");

InfoJury infoJury = new InfoJury(nom,prenom,idJury);

listeJuree.add(infoJury);

}

System.out.println("Liste jurées envoyé");

return listeJuree;

}

//getAllCandidatAdmissible()

public List<Integer> getAllCandidatAdmissible() throws SQLException {

ResultSet rs = this.statement.executeQuery("SELECT fk\_idCandidat FROM Hist\_Statuts WHERE statut='admissible';");

List<Integer> listeCandidatAdmissible= new ArrayList<Integer>();

while (rs.next()) {

int idCanditat = rs.getInt("fk\_idCandidat");

listeCandidatAdmissible.add(idCanditat);

}

return listeCandidatAdmissible;

}