

Énoncé

En prévision des élections des représentants des étudiants dans le conseil scientifique de l'ENIG, le centre informatique souhaite mettre en place une application web JEE basée sur le **pattern DAO** et l'API JPA, gérant les données concernant les électeurs, les candidats et les résultats de votes.

- Un **Électeur** est caractérisé par un identificateur, un nom, un login, un mot de passe, son niveau d'études (1 : ingénieur, 2 : mastère et 3 : doctorant) et son état de vote (vrai : a voté ou faux : pas encore).
- Un **Candidat** est caractérisé par un identificateur, un nom, un prénom et son niveau d'étude.
- Chaque électeur doit s'authentifier pour voter. Il doit choisir un seul candidat, selon son niveau d'étude.
- Le **Vote** est rattaché à un candidat et est caractérisé par un identificateur et le nombre de votes pour chaque candidat.

Étape 1 : Développement de la couche modèle

1. Définir pour chacune des classes : **Electeur**, **Candidat** et **Vote** leurs attributs (déclarés privés) seulement. N'oubliez pas de prendre en considération la relation entre ces classes.

Étape 2 : Développement de la couche dao

Soit l'interface générique **IGestionVote**, cette interface contient la déclaration des méthodes :

1. `Electeur` `verif(String login, String pswd);`
2. `List<Vote> getAllVote();`
3. `List<Candidat> getAllCandidatByNiv(int niv);`
4. `boolean getEtatVote(Electeur e);`
5. `Vote getVoteByCandidat(int idc);`
6. `void addVote(Vote e);`
7. `void updateVote(Vote e);`
8. `void updateEtatVote(Electeur e);`
9. `long getNbElecteur();`

Créer la classe **GestionVoteImp** qui implémente l'interface **IGestionVote** et redéfinissez les méthodes de l'interface.

Étape 3 : Développement de la couche contrôleur et vue

1. L'application est démarrée à partir de la page d'authentification « **login.jsp** ». Une fois l'électeur a saisi son login et son mot de passe et il clique sur le bouton « **Se connecter** », il sera redirigé vers le servlet « **ServAuth.java** ». Cette dernière va vérifier :
 - a. Si les paramètres de connexion sont valides et l'électeur n'a pas encore voté, il sera dirigé vers la page « **vote.jsp** »
 - b. Si les paramètres de connexion sont valides et l'électeur a déjà voté, il sera dirigé vers la page « **resultatsVote.jsp** »
 - c. Si les paramètres de connexion ne sont pas valides, un message d'erreur s'affiche dans la page « **login.jsp** ».
2. Dans la page « **vote.jsp** », la liste des candidats qui correspondent au niveau d'études de l'électeur s'affiche. Ainsi, l'électeur choisi un candidat et il clique sur le bouton « **voter** ». En ce moment, les deux variables état de vote de l'électeur « **etatvote** » et le nombre de votes « **nbvotes** » seront mis à jour dans la BDD via le servlet « **ServVote.java** ».
3. Dans la page « **resultatsVote.jsp** », on affiche le résultat des votes pour chaque candidat (nom des candidats avec leur taux de vote). Avec :

$$taux_{vote} = nombre_vote_candidat / nombre_electeurs * 100$$

Travail demandé : (voir ANNEXE)

1. Implémenter les différentes classes et servlets susmentionnées.
2. Apporter les ajouts nécessaires aux différentes pages Web : « **login.jsp** », « **vote.jsp** » et « **resultatsVote.jsp** » *en utilisant la bibliothèque JSTL.*

ANNEXE

MLD

TABLE `Electeur` (`ide` int(11) NOT NULL, `nom` varchar(45) NOT NULL, `login` varchar(45) NOT NULL, `pswd` varchar(45) NOT NULL, `niveau` int(11) NOT NULL, `etatvote` boolean NOT NULL)

TABLE `Candidat` (`idc` int(11) NOT NULL, `nom` varchar(45) NOT NULL, `prenom` varchar(45) NOT NULL, `niveau` int(11) NOT NULL)

TABLE `Vote` (`id` int(11) NOT NULL, `nbvotes` int(11) NOT NULL, `idc` int(11) NOT NULL)

Page Web "login.jsp"

Login	Login
<p>Login: <input style="width: 80%;" type="text"/></p> <p>Password: <input style="width: 80%;" type="password"/></p> <p style="text-align: center; margin-top: 10px;"><input type="button" value="Se connecter"/></p>	<p>Login: <input style="width: 80%;" type="text"/></p> <p>Password: <input style="width: 80%;" type="password"/></p> <p style="text-align: center; margin-top: 5px;">Invalid login/password</p> <p style="text-align: center; margin-top: 10px;"><input type="button" value="Se connecter"/></p>

Page Web "vote.jsp"

Liste des candidats

☐ Ben Ali Hammadi

☐ Thabti Nizar

Page Web "resultatsVote.jsp"

Résultats des votes

Kadri Mohamed	0%
Rejeb Samir	0%
Ben Ali Hammadi	33,33%
Salah Samia	0%
Thabti Nizar	0%

Code JSP des pages web

« login.jsp »

```

1. <%@ page language="java" contentType="text/html; charset=UTF-
   8"
2. pageEncoding="UTF-8"%>
3. <html> <head> <meta charset="utf-8">
4. <title>Authentification</title>
5. </head>
6. <body>
7. <div style="text-align: center">
8. <h1>Login</h1>
9. <form action="..." method="...">
10.     <input type="text" name="login" size="30" />
11.     <br><br>
12.     <input type="password" name="password" size="30" />
13.     <br> ...
14.     <br><br>
15.     <button type="submit">Login</button>
16. </form> </div></body> </html>

```

« vote.jsp »

```

1. <%@ page language="java" contentType="text/html; charset=UTF-
   8"
2. pageEncoding="UTF-8"%>
3. <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"
   %>
4. <html>
5. <title>Voter</title>
6. <link rel="stylesheet" type="text/css" href="cssf.css">
   </head>
7. <body>

```


@Entity

```
class Ecteur {
```

```
    @Id
```

```
    int id;
```

```
    String nom;
```

```
    String pass;
```

```
    int niv;
```

```
    boolean etat-vote }
```

TD4 :

@Entity

```
class candidat {
```

```
    @Id
```

```
    int idc;
```

```
    String nom;
```

```
    String prenom;
```

```
    int niv ; }
```

@Entity

```
class vote {
```

```
    @Id
```

```
    int id;
```

```
    int mbjo;
```

```
    @Many toOne
```

```
    candidat cd; }
```

Dao :

```
Electeur verif (String login, String pswd) {
```

```
    Query q = em.createQuery("select e from Electeur e  
    where e.getLogin() = :l and e.getPswd() = :p")
```

```
    q.setParameter("l", login);
```

```
    q.setParameter("p", pswd);
```

```
    Electeur e = q.getSingleResult();
```

```
    return e; }
```

```
boolean getEtatVote (Electeur e) {
```

```
    Query q = em.createQuery
```

```
5/ vote getVoteByCandidat (int idc) {
```

```
    Query q = em.createQuery("select v from vote v where v  
    v.getCd().getIdc() = :idc ;
```

```
    vote v = q.getSingleResult();
```

```
    return v; }
```

```
4/ void updateEtatVote (Electeur e) {
```

```
    Electeur e1 = em.find(Electeur.class, e.getId());
```

```
    e1.setEtatVote(true);
```

```
    em.getTransaction().begin();
```

```
    em.merge(e1);
```

```
    em.getTransaction().commit(); }
```



```

long getNbElecteur() {
    Query q = em.createQuery("select count(e.id) from electeur e");
    long nb = (long) em.getSingleResult();
    return nb;
}

```

Servlet

```

class servAuth extends HttpServlet {
    GestionVoteImp gest = new GestionVoteImp();
}

```

```

public doPost(HttpServletRequest Req, HttpServletResponse Res) {
    String login = Req.getParameter("login");
    String pswd = Req.getParameter("pswd");
    Electeur e = Gest.verif(login, pswd);
    if (e != null) and (e.getEtatVote() == false) {
        session.setAttribute("e", e);
        resp.sendRedirect("vote.jsp");
        // list < candidat > lc = Gest.getAllCandidat();
        // By div (e.getId())
        // request.setAttribute("lc", lc);
        // request.getRequestDispatcher("vote.jsp").forward(R, resp);
    } else {
        if (e != null) and (e.getEtatVote() == true) {
            session.setAttribute("e", e);
            resp.sendRedirect("servVote.gawa");
        } else {
            String msg = "Invalid login/password";
            Request.getRequestDispatcher("login.jsp").forward(Req, Res);
        }
    }
}

```

```

public class servVote extends HttpServlet {
    doPost(HttpServletRequest req, HttpServletResponse res) {
        GestionVoteImp gs = new GestionVoteImp();
        int ldc = Integer.parseInt(req.getParameter("ldc"));
        vote v = gs.getVoteByCandidat(ldc);
        v.setNb(v.getNb() + 1);
        gs.updateVote(v);
        gs.updateEtatVote(session.getAttribute("e"));
        list < vote > ls = gs.getAllVote();
        req.setAttribute("lv", ls);
        long mbe = gs.getNbElecteur();
        req.setAttribute("mbe", mbe);
        req.getRequestDispatcher("servResult.html").forward(req, res);
    }
}

```