

	République Tunisienne Ministère de l'Enseignement Supérieur et de la Recherche Scientifique Université de Gabès École Nationale d'Ingénieurs de Gabès	Réf : DE-EX-01
	<u>EPREUVE D'EVALUATION</u>	Indice : 3
		Date : 02/12/2019
		Page : 1/5

Année Universitaire : 2020/2021	Date de l'Examen : 24/11/2021
Nature : <input checked="" type="checkbox"/> DC <input type="checkbox"/> Examen <input type="checkbox"/> DR	Durée : <input type="checkbox"/> 1h <input checked="" type="checkbox"/> 1h30min <input type="checkbox"/> 2h
Diplôme : <input type="checkbox"/> Mastère <input checked="" type="checkbox"/> Ingénieur	Nombre de pages : 05
Section : <input type="checkbox"/> GCP <input type="checkbox"/> GCV <input type="checkbox"/> GEA <input checked="" type="checkbox"/> GCR <input type="checkbox"/> GM	Enseignant : I. Azaiez
Niveau d'étude : <input type="checkbox"/> 1 ^{ère} <input checked="" type="checkbox"/> 2 ^{ème} <input type="checkbox"/> 3 ^{ème} année	Documents Autorisés : <input type="checkbox"/> Oui <input checked="" type="checkbox"/> Non
Matière : Programmation III	Remarque :

NB. La qualité de rédaction, la clarté et la précision des réponses et l'utilisation des commentaires seront des éléments d'évaluation très importants. **Le barème est donné à titre indicatif (/ 40 pts)**

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, 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

(8 pts)

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

(15 pts)

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 seulement les méthodes de l'interface : 1,3,4,5 et 9. Les autres méthodes sont supposées définies.

Étape 3 : Développement de la couche contrôleur et vue

(17 pts)

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 les scriptlets java ou bien la bibliothèque JSTL.

NB. Vous pouvez utiliser l'API JPA pour la couche modèle et dao.

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` (`ide` 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, `ide` int(11) NOT NULL)

Connexions à la BDD

```
Class.forName("com.mysql.jdbc.Driver");  
String url ="jdbc:mysql://localhost:3306/elections";  
Connection con = DriverManager.getConnection(url,"root","root");  
Ou bien
```

```
EntityManagerFactory fact =  
Persistence.createEntityManagerFactory("elections");  
EntityManager em= fact.createEntityManager();
```

Page Web "login.jsp"

Login	Login
Login: <input type="text"/>	Login: <input type="text"/>
Password: <input type="password"/>	Password: <input type="password"/>
<input type="button" value="Se connecter"/>	Invalid login/password <input type="button" value="Se connecter"/>

Page Web "vote.jsp"

Liste des candidats

☐ Ben Ali Hammadi

☐ Thabti Nizar

Résultats des votes	
Kadri Mohamed	0%
Rejeb Samir	0%
Ben Ali Hammadi	33,33%
Salah Samia	0%
Thabti Nizar	0%
<input type="button" value="logout"/>	

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. `

`
12. `<input type="password" name="password" size="30" />`
13. `
 ...`
14. `

`
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>`
8. `<form action="..." method="..."> <fieldset>`

« ens.jsp »

1. <table class="table table-hover">
2. <tr><th>ID</th> <th>Nom</th><th>Fonction</th>
3. <th>Nbre Heures Cours</th> <th>Nbre Heures TP</th>
<th>ACTION</th></tr>
4. <tbody>
5.
6. </tbody>
7. </table>

« surv.jsp »

1. <table class="table table-hover">
2. <tr><th>ID</th> <th>Matière</th><th>Jour</th> <th>Salle</th>
3. <th>Groupe</th> </tr>
4. <tbody>
5.
6. </tbody>
7. </table>

« affect.jsp »

1. <table class="table table-hover">
2. <tr><th>ID</th> <th>Enseignant</th><th>Matière</th>
<th>Salle</th> <th>Jour</th> <th>Groupe</th> </tr>
3. <tbody>
4.
5. </tbody>
6. </table>