```
Public class Electeur {
    private
    @Id
    @GeneratedValue (Strategy = GenerationType.Identity)
    private int id;
    private String nom;
    private String login;
    private String mp;
    privat int nivetd;
    private boolean etat_vote. }
@Entity
Public class condidat {
    @Id
    @GeneratedValue (Strategy = GenerationType.Identity)
    private int idc;
    private String nom
            String pre
            int nivetd }

@Entity
Public class vote {
    @Id
    @GeneratedValue (Strategy = GenerationType.Identity)
    private int idv
            int nbrvote

    @ManyToOne
    Condidat c; }
11) Electeur public class GestionVoteImp implements IGestionVote {
        EntityManager em = factory.EntityManagerFactory();
        EntityManager em = factory.createEntityManager.
```

```
Electeur verif (String login, String pswd) {
  Query q = ("select e from Electeur e where e.login = :l , e.psw = :p")
    q.setParammeter ("l", login);
    q.setParammeter ("psw", psw);
  Electeur e = q.getSingleResult ()
return e ; }
2) List<vote> getALLvote() {
    Query q = ("select v from vote v")
  List<vote> Lv = q.getResultList() ;
return Lv; }
3) List<condidat> getALLCondidatByNiv (int niv) {
Query q = ("select c from Condidat c where c.niv = :niv")
    q.setParammeter ("niv", niv);
  List<condidat> Lc = q.getResultList();
return Lc; }
4) boolean getEtatvote (Electeur e) {
    boolean Etat = false ;
Query q = ("select e from Electeur ...
    find
  Electeur e = em.find (Etudiant.class e)
  if (e.getEtat-vote().equals (Etat))
      Etat = false )
   else
     Etat = true
return eta.vot
```

Model :

```
@Entity
Public electeur {
    @Id
    @GeneratedValue (Strategy = GenerationType.Identity)
    private int ide ;
    private String nom ;
    private String login;
    private String pass ;
    private int niv ;
    private boolean etat_vote ; }

@Entity
Public Condidat {
    @Id
    @Generated value (Strategy = GenerationType.Identity)
    private int idc ;
    private String nom;
    private String prenom ;
    private int niv ;  }

@Entity
Public vote {
    @Id
    @ GeneratedValue( Strategy = Generation Type. Identity)
    private int idv;
    private int nbnv ;
    private
@ Many-to one
private Condidat c ;
```

```
public GestionVoteImp implements IGestionVote {
    EntityManagerFactory emf = Entity Factory create
    EntityManager em = factory.createEntityManager ( );

    Electeur verif (String Login , String pswd ) {
    Query q = em.createQuery ("Select e from Electeur e where
    e.login = :Login, e.pswd = :pass ") ;
        q.getParammeter ("Login", login) ;
        q. getParammeter ("pass", pswd) ;
    Electeur e = q.getSingleResult( );
    return e; }
List<vote> getALLvote ( ) {
    List<vote> = em.find
Query q = em.createQuery ("Select * from vote v "),
    q. getResultList ( )
List<vote> le =
    return le. }
```

```
List<Candidat> getAllCandidatByNiv(int niv)
Query q = em.createQuery("Select c from Candidat c
  where c.getniv() = :n");
q.getAttribute
q.getParammeter("n", niv);
List<Candidat> lc = q.getResultList();
return lc; }

boolean getEtatVote(Electeur e){
   boolean Etat_vote = false
Query q = em.CreateQuery("Select e from Electeur e where
etat-vote = :vote");
q.getParammeta("vote", e.getEtat_vote());
boolean
Etat_vote() = Parse.Integer.q
   if(e.getEtat_vote().equals(Etat_vote))
       Etat_vote = true;
turn Etat_vote;
```

```
Vote getVoteByCandidat(int ide){
Query q = em.createQuery("Select v from vote v where
v.getId().getide() = :ide");
   q.getParammeter("ide", ide);
Vote v = q.getSingleResult()
return v; }
void addVote(Vote e){
    em.getTransaction.begin();
    em.persist(e);
    em.getTransaction.commit(); }
void updateVote(Vote e){
    em.getTransaction().begin();
Vote e1 = em.find(Vote.class, e);
    em.merge(e1);
    em.getTransaction().commit(); }
void updateEtatVote(Electeur e){
Electeur e1 = em.Find(Vote.class, e)
    e1.getEtat_vote(true);
    em.getTransaction().begin();
    em.merge(e1)
em.getTransaction().commit();
```

```
servote extends HttpServlet {

boolean etatvote = ParseBoolean.getParameter("etatvote");
int nbvote = ParseInteger.getParameter("nbv");
if(etatvote ==

int ide = IntegerParseInt(gst.getvoteByCandidat
```

```
servote extends HTTPServlet {
  int ide = Integer.ParseInt(request.getParameter("ide"))
  vote v = gst.getvoteByCandidat("idc")
v.setNb(v.getnbvote + 1); gs.update vote(v);
   gs.update Etat_vote (session.getAttribute(e));
     request.setAttribute("v",v)
```

```
int ide = Integer.ParseInt(request.getParameter("ide"))
vote v = gst.getVoteByCandidat(ide)
   v.setNb(v.getNbvote + 1);
    gs.update vote(v);
    gs.update Etat vote(v);
List<vote> lv = gs.getALL vote();
    request.setAttribute("Lv", Lv);
long nbv = gs.getNbElecteur();
   request.setAttribute("nbv", nbv);
request.getRequestDispatcher("Result.jsp").forward(req,resp)
```

```
login :
<body>
<form action="servAuth" method="Get">
<br>
  <c:out value = "${"invalide"}"></c:out
```

```
Long getNbElecteur(){
Query q = em.createQuery("select count(e.idle) from
Electeur")
Long nbElecteur = (Long) em.getSingleResult();
return nbElecteur;}
```

Contrôleur :

```
public servAuth extends HTTPServlet{
  GestionVoteImp gst = new gestionVoteImp();
doGet(HTTPRequest request, HTTPResponse response){
String login = request.getParameter("login");
String pass = request.getParameter("pass");
HttpSession session = request.getSession();
Electeur e = gst.verif(login, pass);
if((e!=NULL) and (e.getEtatVote()=false))
    session.setAttribute("e", e);
List<Condidat> Lc = gst.getAllCondidatByNiv(e.getNiv());
  (e.getNiv());
request.setAtt("Lc", Lc)
request.getRequestDispatcher("vote.jsp"),
resp.sendRedirect("vote.jsp");
```

```
servAuth extends HTTPServlet{
doGet(request, response){
GestionVoteImp gst = new GestionVoteImp();
String login = request.getParameter("login");
String pass = request.getParameter("pass");
Electeur e = gst.verif(login, pass);
if(e!=NULL And gst.getEtatVote(e)==false)
    session.setAttribute("e",e);
List<Condidat> Lc = gst.getAllCondidatByNiv(e.getniv());
  request.setAttribute("Lc", Lc);
  requestgetrequestDispatcher("vote.jsp").forward(
request, response);
  response.sendRedirect("vote.jsp");
else
  if(e!=NULL and gst.getEtatvote(e)==true)
    session.setAttribute("e",e);
List<Condidat>Lc = gst.getALLCondidatByNiv(e.getniv());
  request.setAttribute("Lc",Lc);
  request.getRequestDispatcher("resultatsVote.jsp").
  forward(request,response);
  response.sendRedirect("resultatsvote.jsp");
else
out.print("invalid login/pass");
  msg = "Invalid login/passwd"
request.getRequestDispatcher("Login.jsp").forward(Req,Rq)
```

JSP Recap :

choose = switch case

test : teste le contenu

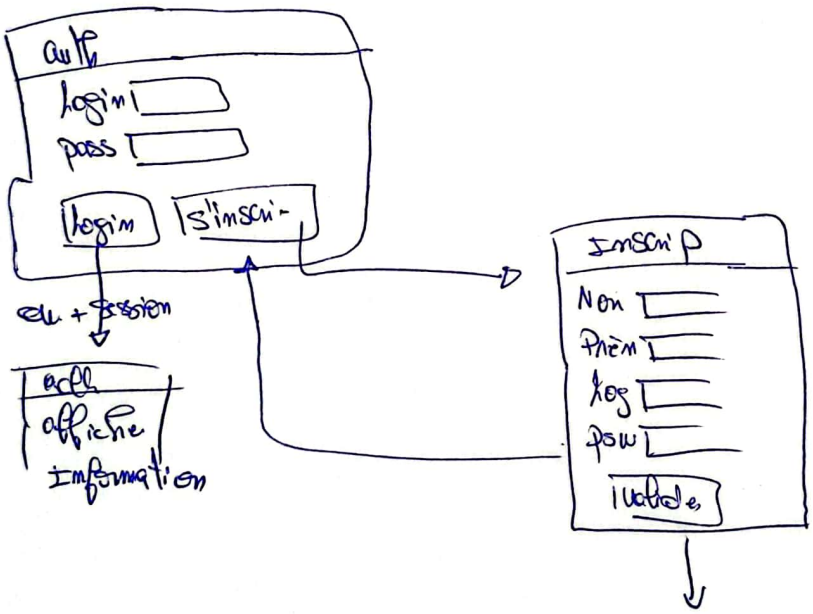
forEach :

jsp :

```
<% List<P> lp = request.getAttribute("lp"); %>
<% for(int i = 0; i < l.size(); i++) { %>
<tr>
<td> <% l.get(i).getNom(); %> </td>
<td> <% l.get(i).getPrenom(); %> </td>
<td> <a href="servl ? id = <% = i %>"> <% = l.get(i).getId() %> </td>
</tr>
<% } %>
```

JSTL

```
<c:forEach = "${lp}" var = "p">
<td>                                   var status = index
${p.Nom}</td>
<td> ${p.prenom} </td>
<td> <a href = "servl ?id = index"> delete </td>
</tr>                                  = ${p.id}"> .. .. .
</c:forEach>
```



@ Entity

User { int id

String login ;

String pswd ;

String role ;

}

Authentification

dao :

verif

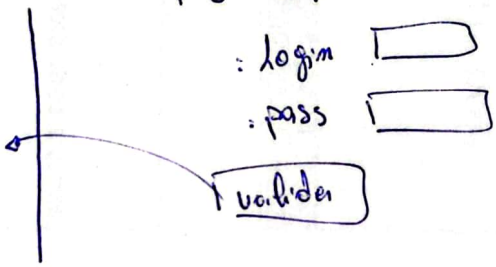
servlet :

class servAuth ext —

User u = verifier (login , pass)

↳ recuperation du Role.

u.getRole ()



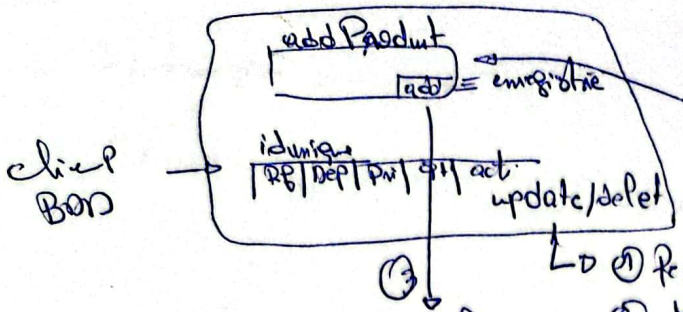
Jsp :

<c:if test

↙ si la reponse .

JSP :

```
<form action = "servletP" methode = "Post" >
    <input type = "text" name = "titre" value = "${s.titre}" >
    <select name = "Etudiant">
     <c:foreach items = "${le}" var = "p">
        <option value = "${p.ide}" > ${p.nom} </option>
     </c:foreach>
    </select>
    <input Date
    <select
</table>
<th> titre </th> <th> Etud </th> ....
  <c:forEach items = "${ls}" var = "s" >
    <tr>
    <td> ${s.titre} </td>
    <td> ${s.e.nom} </td>
    :
    </tr>
  </c:foreach>
```



Resultat ① ②

① Recherche → servlet recherche

② Mise à jour → servlet (update)

Appel DAO

getProduitById (int id)

produit

produit p = ...

(a)

d) Produit P = request.getAttribute ("p")

servlet (update)

DAO update / Add

If (getProduitById (id) == null)
add
else
update.

(b) request.setAttribute ("p", "p");
"lp"
↳ ArrayList

RequestDispatcher : pour envoyer request : redirect

response.setRedirect ("serv2 ? x = 2");

partage entre 2

Index.jsp

① page web .jsp

1.1 - Demande de mise à jour (clique sur lien Hyp Appartin du tab)
2 - Redirection vers servlet Recherche →
3 - Appel de la méthode getProduitById (id) →
4 - redirection vers index.jsp { request.setAttribute ("p", P);
request.getRequestDispatch (index.jsp").
forward (Request, Reponce)
5 - (index.jsp) Modifier les champs "sauf produit" + il clique sur enregistrer
6 - Redirection vers servlet 1 : traitement
Actions formulaire 7 - Récupération du champs au niv serv 2 (Request.getParameter)
8 - teste si Id existe → update Produit P = request.getAtt ("p");
update
sinon add (P).
9 - get ALL produit → lp
10 - request.setAtt ("Lp", lp); redirection index.jsp.

Collaboration entre les servlets : Mohamed Youssfi

Serv1 ⟷ Serv2

Serv1 → page.web.jsp
↑
Serv2

liste de Produit : lp → serv2

object obj ----> serv2 ---> pageweb

en bref

Stocker obj ds request

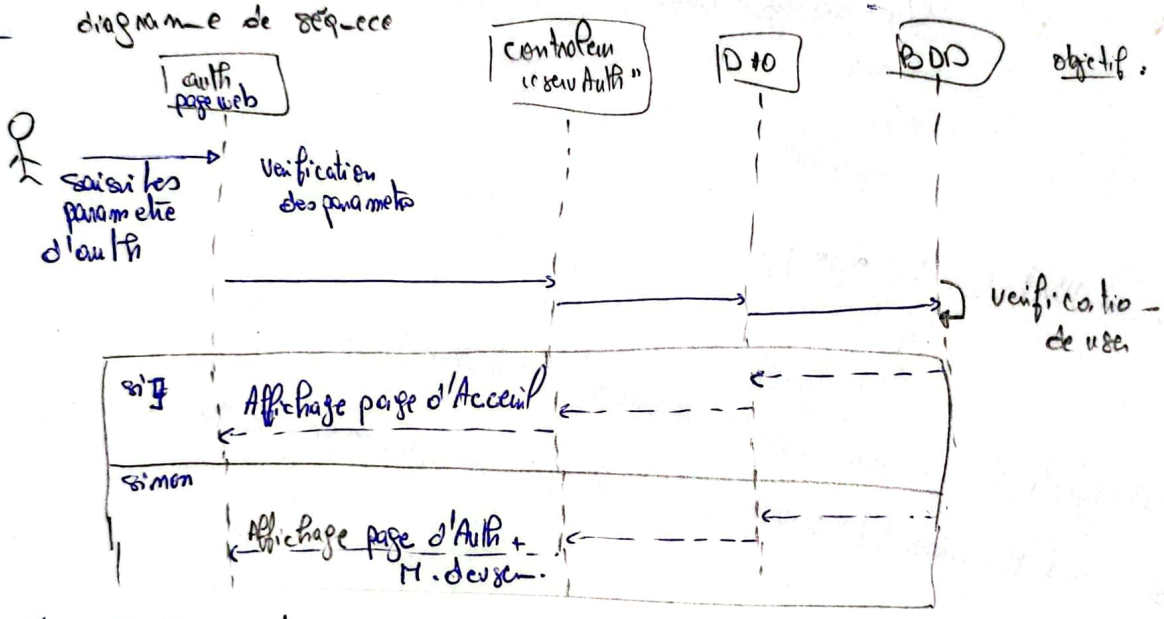
"redirection etde d

=> request.setAttribute("obj", obj);

request.getRq disp("serv2").forward(Rq, Resp);

STS
"
eclipse.

6g diagramme de séquence



Les étape d'auth en moti— du saisissen :

① saisir les paramètres de connexion (login, pass)
② création d'un objet user u ;
③ verifier si u est dans la saisie
Lo sinon si u ∃ dans BBD => getSession() : notNULL
Lo si oui sauvgarder u dans la saisie
Lo sinon erreur
Lo si oui Aller à la page suivante

couche Model.

class user {
String login,
" pass ;
;
}

Html :

```
<form action = "verifSev" methode = "Get" >
user name < input type = "text" name = "name" />
pass < input type = "text" name = "pass" />
<input type = "submit" value = "login" />
</form>
```

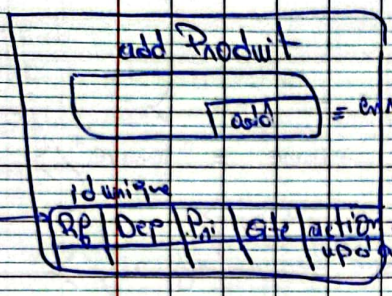
verifSev :

```
doGet ( ___ ) {
    String name = request.getParammeter ("name");
    String pass = request.getParammeter ("pass");

    if ( name.equal("admin") ) {
        write.println ("valid user");
    }
    else
        write.println ("use not Exist");
```

printwriter write = responce.getwriter();

together forever



1_ demande de mise à jour (cliquer sur lien hypertext)

2_ ↳ Redirection vers servlet Recherche

↳ Appel de la méthode getProduitById(id);

→ redirection vers index.jsp ; req.setAttribute("P",P);
req.getRequestDisp(index.jsp").
forward(Request Reponce)

3_ . modifier les champs puis cliquer sur enregistrer.

↳ redirect vers servlet Principal 1.

4/ ↳ Récupération du champ au niv servlet (Request.getParameter(.));

↳ teste si id ∃ → ProduitP = request.getAtt("P");
update
sinon add(P).

5/ → getAllProduit → lp

↳ req.setAtt("lp", lp)

↳ redirection vers index.jsp

together forever