



# La banque en ligne

# Introduction

# Objectifs du projet

- Gestion de comptes bancaires en ligne
- Transactions entre les comptes
- Filtres de recherche
- Partie administrateur

# Technologies utilisées

- Notions connues :

- Java (CPOA)
- Base de données
- Gestion de projet
- Pages web HTML/CSS

- Nouvelles notions :

- Application web
- Java (JEE)
- Serveur Tomcat

# Partie gestion de projet

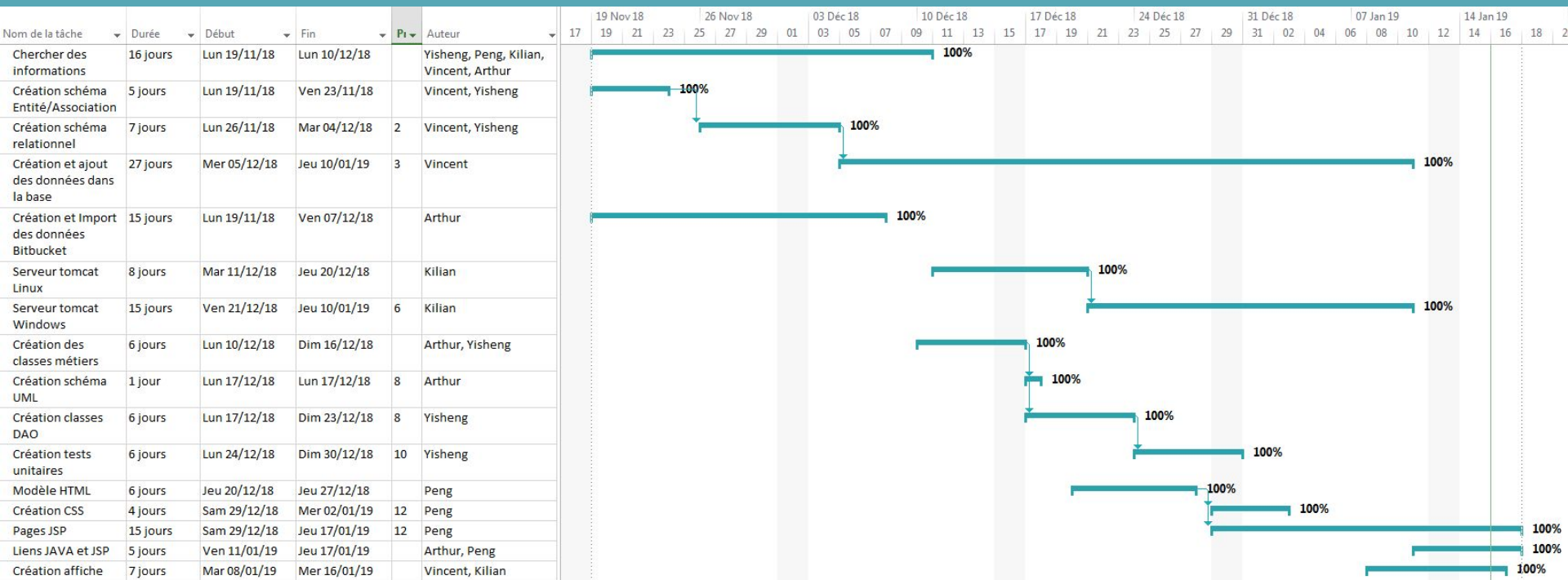
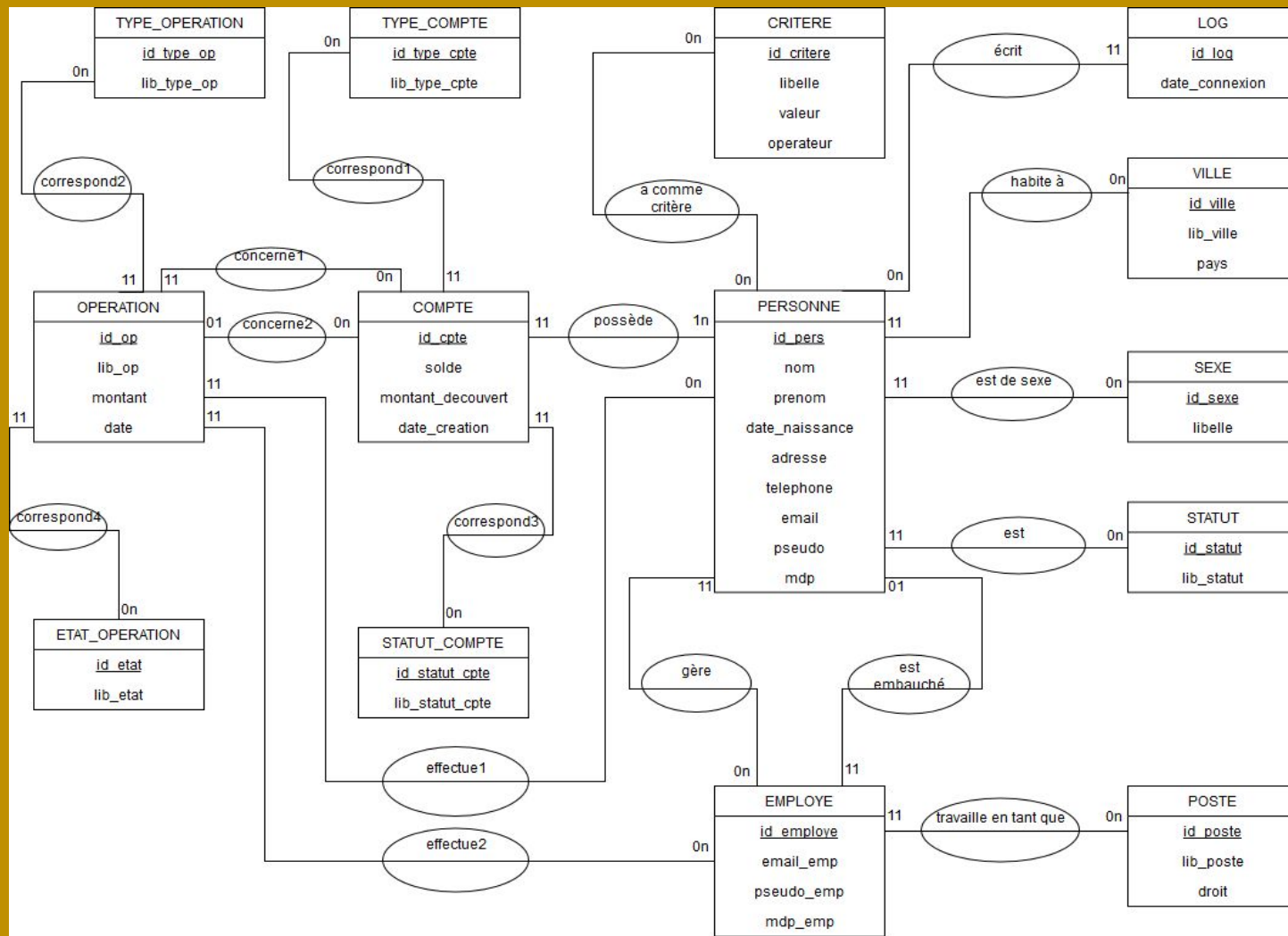


Diagramme de Gantt

**Partie base de  
données**

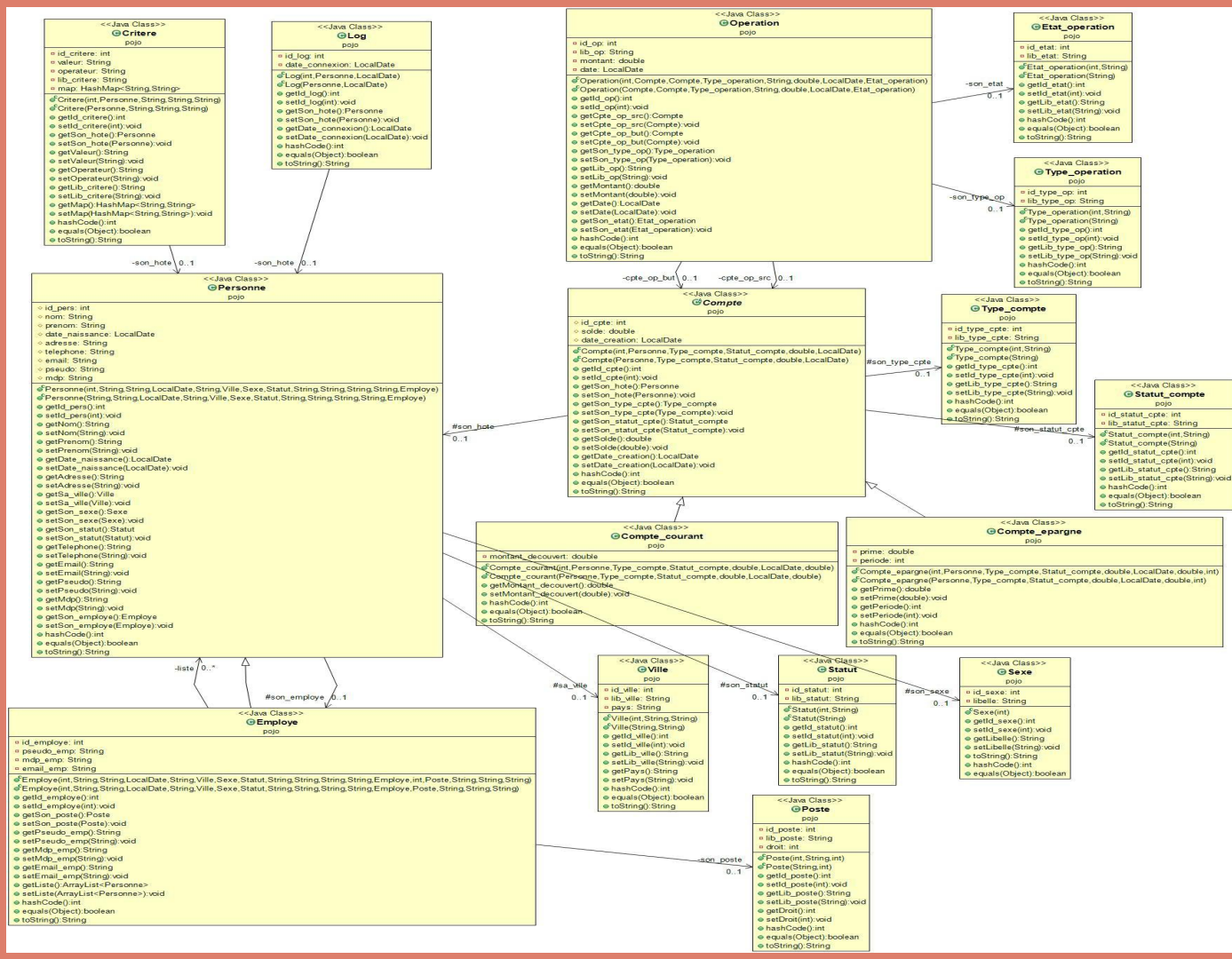
# Schéma Entité/ Association





# Partie Java

# Diagramme des classes



# Back-end : Les classes DAO

```
@Override
public int create(Compte x1) {
    String req = "insert into COMPTE (id_pers, id_type_cpte, id_statut_cpte, solde, montant_decouvert"
        + ", date_creation)" + " values(?,?,?,?,?,?)";

    try {
        this.reqprep = SQLconnexion.getInstance().creeConnexion().prepareStatement(req,
            Statement.RETURN_GENERATED_KEYS);
        this.reqprep.setInt(1, x1.getSon_hote().getId_pers());
        this.reqprep.setInt(2, x1.getSon_type_cpte().getId_type_cpte());
        this.reqprep.setInt(3, x1.getSon_statut_cpte().getId_statut_cpte());
        this.reqprep.setDouble(4, x1.getSolde());
        if (x1.getSon_type_cpte().getId_type_cpte() == 1) { // compte courant
            Compte_courant cc = (Compte_courant) x1;
            System.out.println("ying" + cc.getMontant_decouvert());
            this.reqprep.setDouble(5, cc.getMontant_decouvert());
        } else { // compte epargne ou d'autres
            Compte_epargne ce = (Compte_epargne) x1;
            this.reqprep.setDouble(5, 0);
        }
        this.reqprep.setDate(6, java.sql.Date.valueOf(x1.getDate_creation()));
        this.reqprep.executeUpdate();
        ResultSet resset = this.reqprep.getGeneratedKeys();
        while (resset.next()) {
            System.out.println("Ajoute");
            return resset.getInt(1);
        }
    } catch (SQLException sqle) {
        System.out.println("SQL Syntaxe Erreur.");
        System.out.println(sqle.getMessage());
    }
    return -1;
}

@Override
public void update(Compte x1) {
    String req = "Update COMPTE set id_pers=?, id_type_cpte=?, id_statut_cpte=?, solde=?"
        + ",montant_decouvert=?, date_creation=? where id_cpte=?";

    try {
        this.reqprep = SQLconnexion.getInstance().creeConnexion().prepareStatement(req);
        this.reqprep.setInt(1, x1.getSon_hote().getId_pers());
        this.reqprep.setInt(2, x1.getSon_type_cpte().getId_type_cpte());
        this.reqprep.setInt(3, x1.getSon_statut_cpte().getId_statut_cpte());
        this.reqprep.setDouble(4, x1.getSolde());
```

# Back-end : Tests Unitaires

The screenshot displays an IDE interface with the following components:

- Project Explorer (Left):** Shows the project structure for 'Banque [projet\_tutoré master]'. The 'test' package is expanded, listing various test classes including 'CompteDAOTest.java', 'CritereDAOTest.java', 'EmployeDAOTest.java', and others.
- JUnit Runner (Top Center):** Indicates the tests finished after 8.024 seconds. Summary statistics show 78/78 runs, 0 errors, and 0 failures, accompanied by a green progress bar.
- Test Results (Center):** A detailed list of test outcomes. Tests are grouped by class, such as 'test.CompteDAOTest' and 'test.EmployeDAOTest'. Each test method is listed with its execution time in seconds (e.g., 'testFindAll (0.705 s)').
- Console (Bottom):** Shows the command prompt output, starting with '<terminated> Test [JUnit] C:\Program Files\Java\jre1.8.0\_181\bin\javaw.exe (2019年1月17日 下午4:07:17)'. Below this, a list of actions is displayed: 'Supprime', 'Ajoute', and 'Disconnected'.



**Partie JSP**

# Contextualisation : JSP et Servlet

## 1) Qu'est-ce qu'une JSP ?

JSP : c'est une page web mélangeant du code HTML et code Java. La page est affichée à l'aide d'un serveur Tomcat.

## 2) Qu'est-ce qu'une classe Servlet, et à quoi elle sert ?

Servlet : classe Java étant liée à une (ou plusieurs) JSP permettant de transmettre des variables aux JSP, ou de récupérer les données saisies par l'utilisateur dans la page JSP, et d'interagir avec la base de données (elles sont liées grâce à une classe au format .xml).





# Classe Servlet et liaison Servlet - JSP

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    request.setCharacterEncoding("UTF-8");
    response.setContentType("text/html; charset=UTF-8");
    String nom= request.getParameter("nom");
    int id_cpte_but = Integer.parseInt(request.getParameter("id_compte_but"));
    double montant = Integer.parseInt(request.getParameter("montant"));

    try {
        HttpSession session = request.getSession(false);
        int id_compte = (int) session.getAttribute("idcompte");
        session.setAttribute("idcompte", id_compte);
        if (session != null) {
            Compte compte_but = CompteDAO.getInstance().getById(id_cpte_but);
            String nom_hote=compte_but.getSon_hote().getNom()+" "+compte_but.getSon_hote().getPrenom();
            if(nom_hote.equalsIgnoreCase(nom)) {
                boolean virement = OperationDAO.getInstance().virement(id_compte, id_cpte_but, montant);
                if(virement ==true) {
                    PrintWriter out = response.getWriter();
                    out.flush();
                    out.println("<script>");
                    out.println("alert('Virement reussi')");
                    out.println("window.location.href='virement_client.jsp'");
                    out.println("</script>");
                }
            }
            }else {

        }

        PrintWriter out = response.getWriter();
        out.flush();
        out.println("<script>");
        out.println("alert('Virement')");
        out.println("window.location.href='virement_client.jsp'");
        out.println("</script>");
    }

    catch(Exception e){
        e.printStackTrace();
    }
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    <display-name>Banque</display-name>
    <welcome-file-list>
        <welcome-file>index.html</welcome-file>
        <welcome-file>index.htm</welcome-file>
        <welcome-file>index.jsp</welcome-file>
        <welcome-file>default.html</welcome-file>
        <welcome-file>default.htm</welcome-file>
        <welcome-file>default.jsp</welcome-file>
    </welcome-file-list>
    <servlet>
        <servlet-name>LoginServlet</servlet-name>
        <servlet-class>web.LoginServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>LoginServlet</servlet-name>
        <url-pattern>/Login</url-pattern>
    </servlet-mapping>
    <servlet>
        <servlet-name>InscriptionServlet</servlet-name>
        <servlet-class>web.InscriptionServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>InscriptionServlet</servlet-name>
        <url-pattern>/Inscription</url-pattern>
    </servlet-mapping>
    <servlet>
        <servlet-name>LogoutServlet</servlet-name>
        <servlet-class>web.LogoutServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>LogoutServlet</servlet-name>
        <url-pattern>/logout</url-pattern>
    </servlet-mapping>
    <servlet>
        <servlet-name>ComfirmerVirementServlet</servlet-name>
        <servlet-class>web.ComfirmerVirementServlet</servlet-class>
    </servlet>
</web-app>
```






# Utilisation


login.jspaccueil\_client.jspLoginServlet.javaModificationServlet.javaAccueilaccueil.jsp

http://localhost:8090/Banque/accueil.jsp

ParticuliersBanque privéeProfessionnelsEntreprisesAssociations

 **AEDI Banque**

S'inscrireSe connecter



## BIENVENUE SUR MABANQUE

Des solutions adaptées pour tous besoins : gestion et  
ouverture de compte en ligne, simulation de crédit, assurance,

Jusqu'au 2 Janvier 2019

**Cartes bancaires :**  
**-50 % pendant 1 an\***

Jusqu'au 31 Janvier 2018

**Bourse Trackers :**  
**0 € frais de courtage**

Jusqu'au 31 Janvier 2019

**Prêt Divers :**  
**1,99 % TAEG fixe**

login.jsp

accueil\_client.jsp

LoginServlet.java


ModificationServlet.java


Inscription


accueil.jsp


http://localhost:8090/Banque/inscription.jsp

# Inscription


 FU


 Yisheng


 10/11/1998


 Ile du Saulcy

Metz

 1111111111

 111111@gmail.com

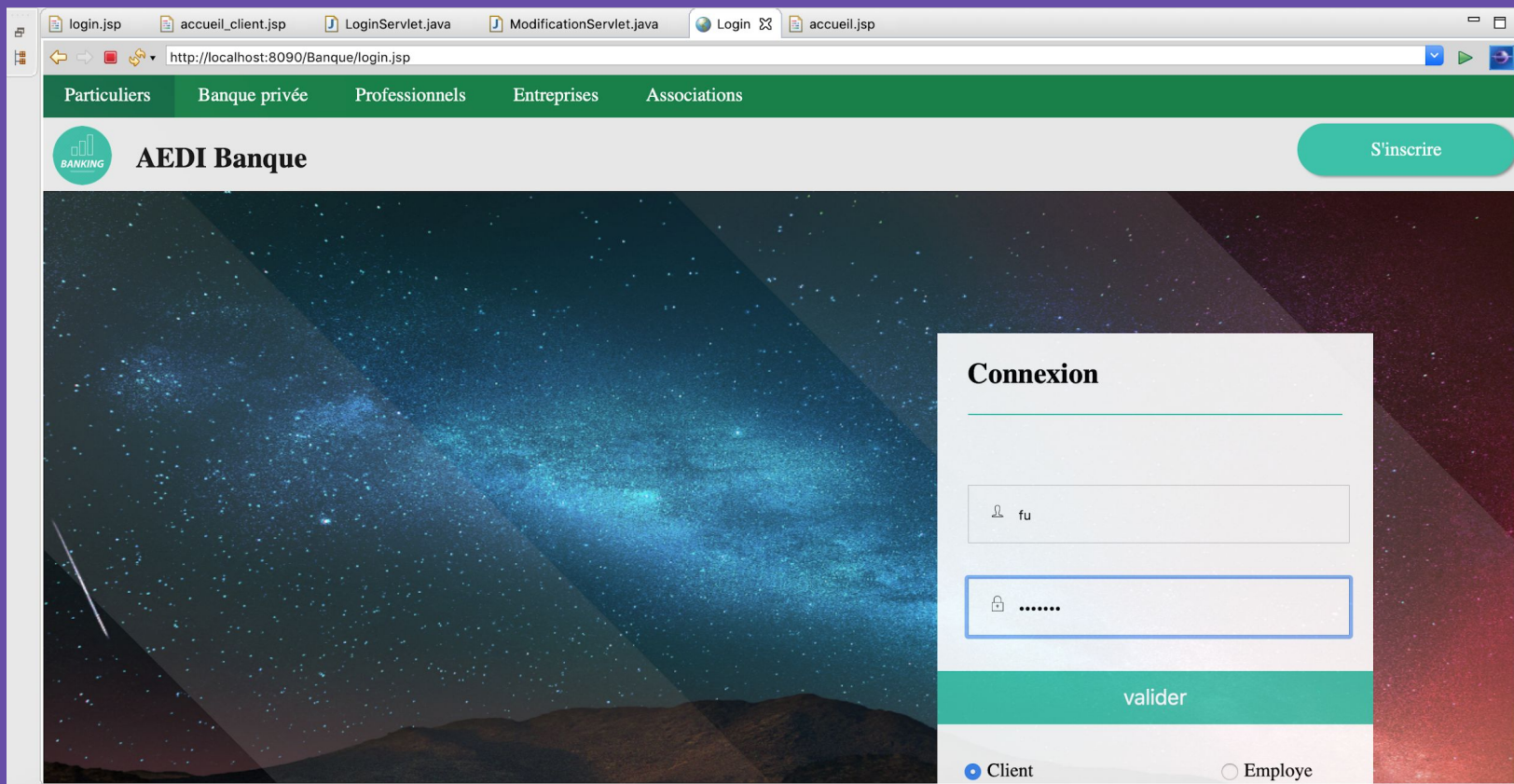
 fu

 .....

Étudiant

☒ Homme


☐ Femme



login.jspaccueil\_client.jspLoginServlet.javaModificationServlet.javaBienvenue

http://localhost:8090/Banque/accueil\_client.jsp


ParticuliersBanque privéeProfessionnelsEntreprisesAssociations

AEDI Banque

Créer un compte

# Bienvenue, *Yisheng FU*

Libelle	Id Compte	Solde
Courant	10007	0.0
Épargne	10008	0.0






login.jspaccueil\_client.jspLoginServlet.javaModificationServlet.javaCompte

http://localhost:8090/Banque/compte\_client.jsp

ParticuliersBanque privéeProfessionnelsEntreprisesAssociations

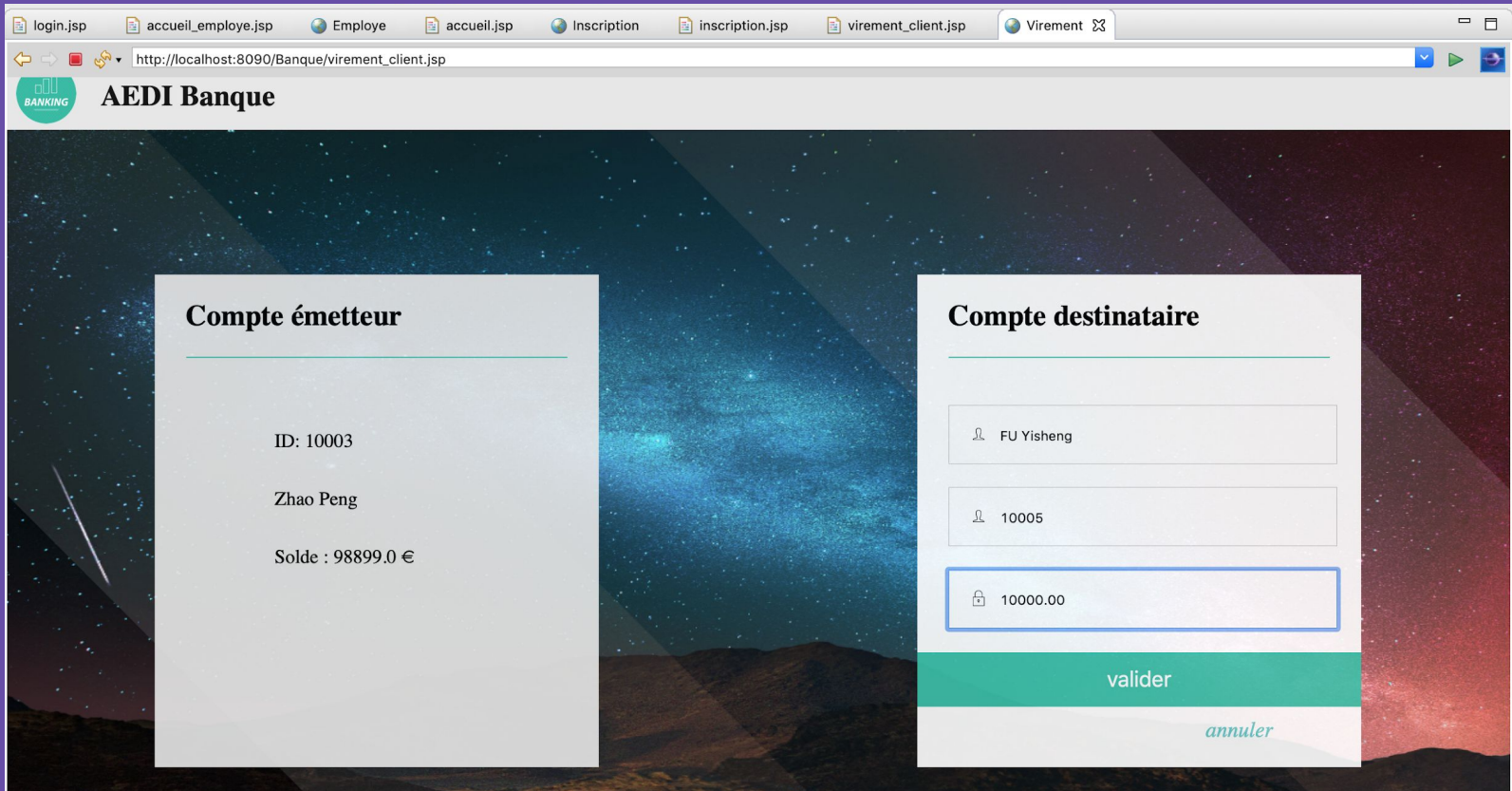
AEDI Banque

Se déconnecter

**Binvenue, Peng Zhao**

Votre solde est de : 98899.0

Date	Compte Source	Compte But	Montant	Libelle d'operation	Etat d'operation
2015-09-16	10003	10001	100.0	virement de 100 euros de compte courant 1 au compte d'épargne 2 2018-12-17 00:00:00	effectué
2017-09-19	10004	10003	100.0	virement de 100 euros de compte courant 1 au compte d'épargne 2 2018-12-17 00:00:00	effectué
2019-01-16	10001	10003	100.0	Virement de 100.0 euros de compte 10003 au compte 100012019-01-16	en cours
2019-01-16	10001	10003	1000.0	Virement de 1000.0 euros de compte 10003 au compte 100012019-01-16	en cours



## Ce qui n'a pas pu être ajouté

- Filtrage des opérations
- Gestion des clients par un employé
- Aspect sécurité



# Conclusion

- Projet enrichissant avec un apport de nouvelles notions
- Travail en groupe
- Temps supplémentaire estimé pour finir le projet : environ 1 mois ½

**Merci de  
votre  
attention !**