

Morgan Lucas

The screenshot shows a Java-based web application interface. At the top, there is a navigation bar with links for 'Home', 'Association', 'Farmeur', 'Ressource', 'Gestion des admins', and a 'Déconnexion' button. Below the navigation bar, there is a 'Connexion' (Connection) form with fields for 'Identifiant' (Identifier) and 'Mot de passe' (Password), and a 'Se connecter' (Connect) button. To the right of the form, large text reads 'JAVA DATABASE ROK'. Below the connection form, there is a section titled 'Vue utilisateur' (User View) with a 'Association' tab selected. This tab displays a table with columns 'Nom ↑↓', 'Effet ↑↓', 'Ress ↑↓', and 'Quantité ↑↓'. The data in the table is as follows:

Nom ↑↓	Effet ↑↓	Ress ↑↓	Quantité ↑↓
Jeanne d'arc	speedwalking	pierre	906
Jeanne d'arc	speedwalking	bois	857
Constance	capacity	terre	906
Constance	capacity	pierre	906
Caius Marius	speedmining	pierre	906

MINI-PROJET

CONTENU

- 
- 
- 03** CONTEXTE
 - 04-06** DIAGRAMME UML
 - 07-08** LES CLASSES
 - 09** LA COUCHE PHYSIQUE
 - 10** LE BEAN ET SON XHTML
 - 11-13** MENTIONS HONORABLES
 - 14** PROBLÈMES RENCONTRÉ
 - 15** CONCLUSION

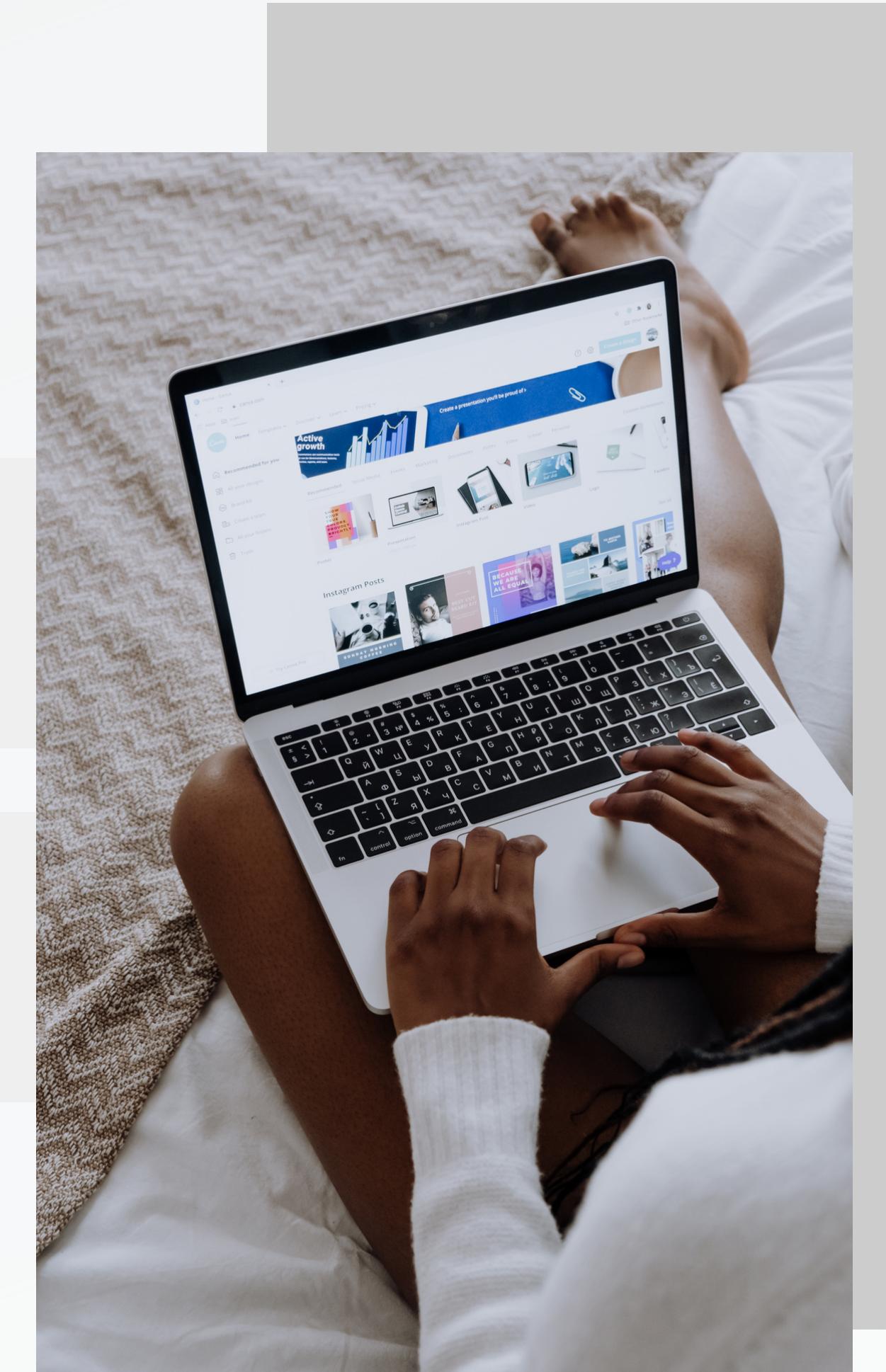
CONTEXTE



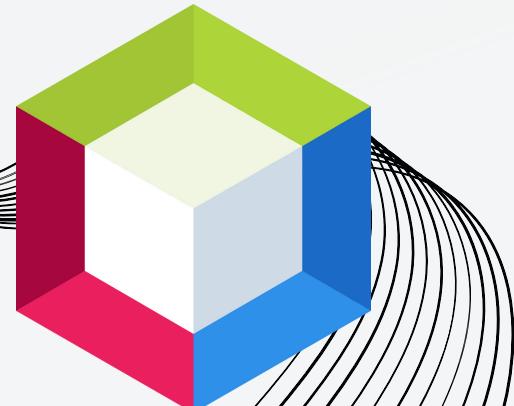
Faire une base de donnée en langage JAVA/XHTML et lui faire une UI



Elle comprendra des **Farmeurs**, des **Ressources** et un **Maire** (admin).



TECHNILOGIE



Maven™



JPA
Java Persistence API

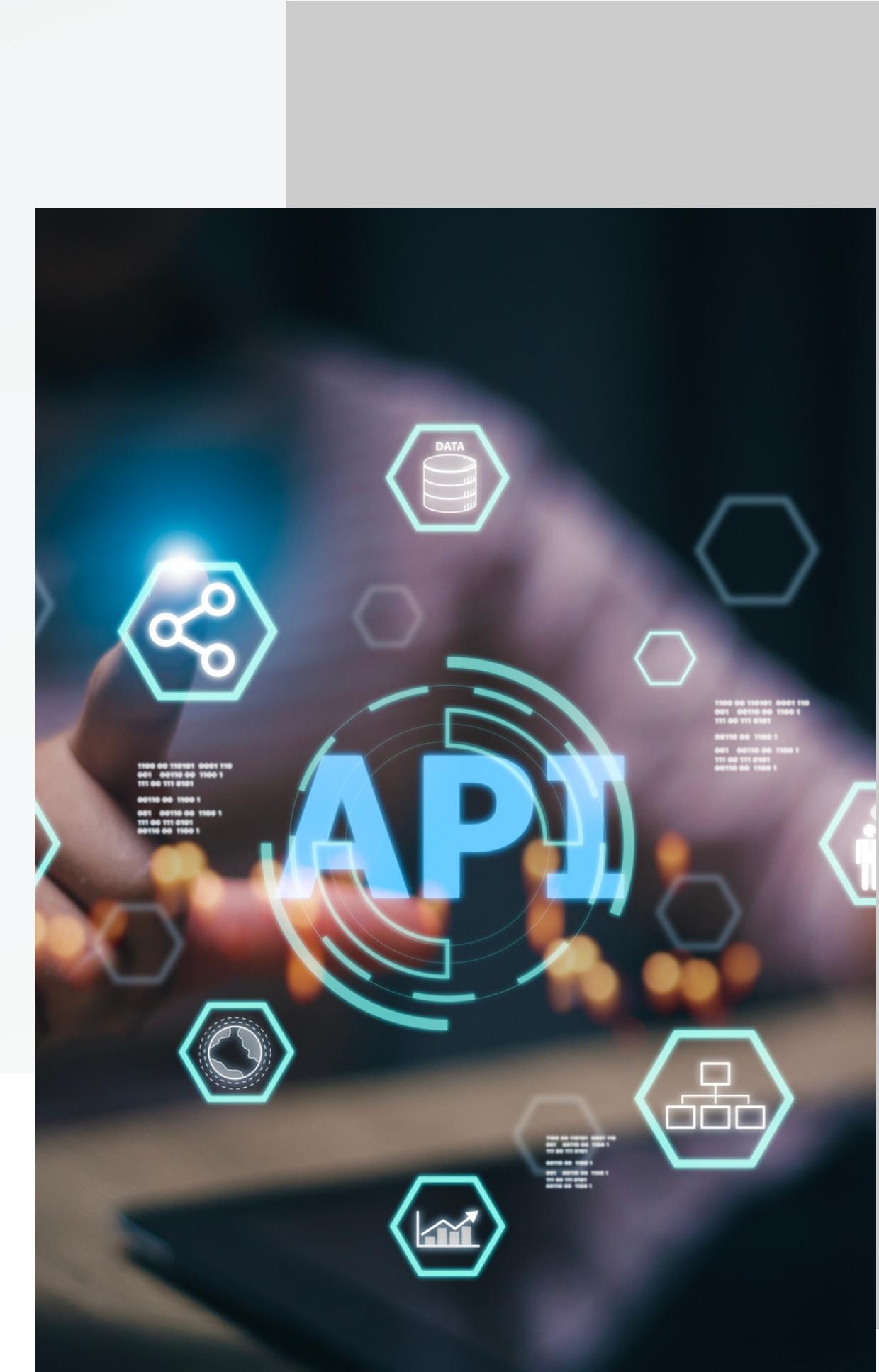


DIAGRAMME UML

Les cas d'utilisation

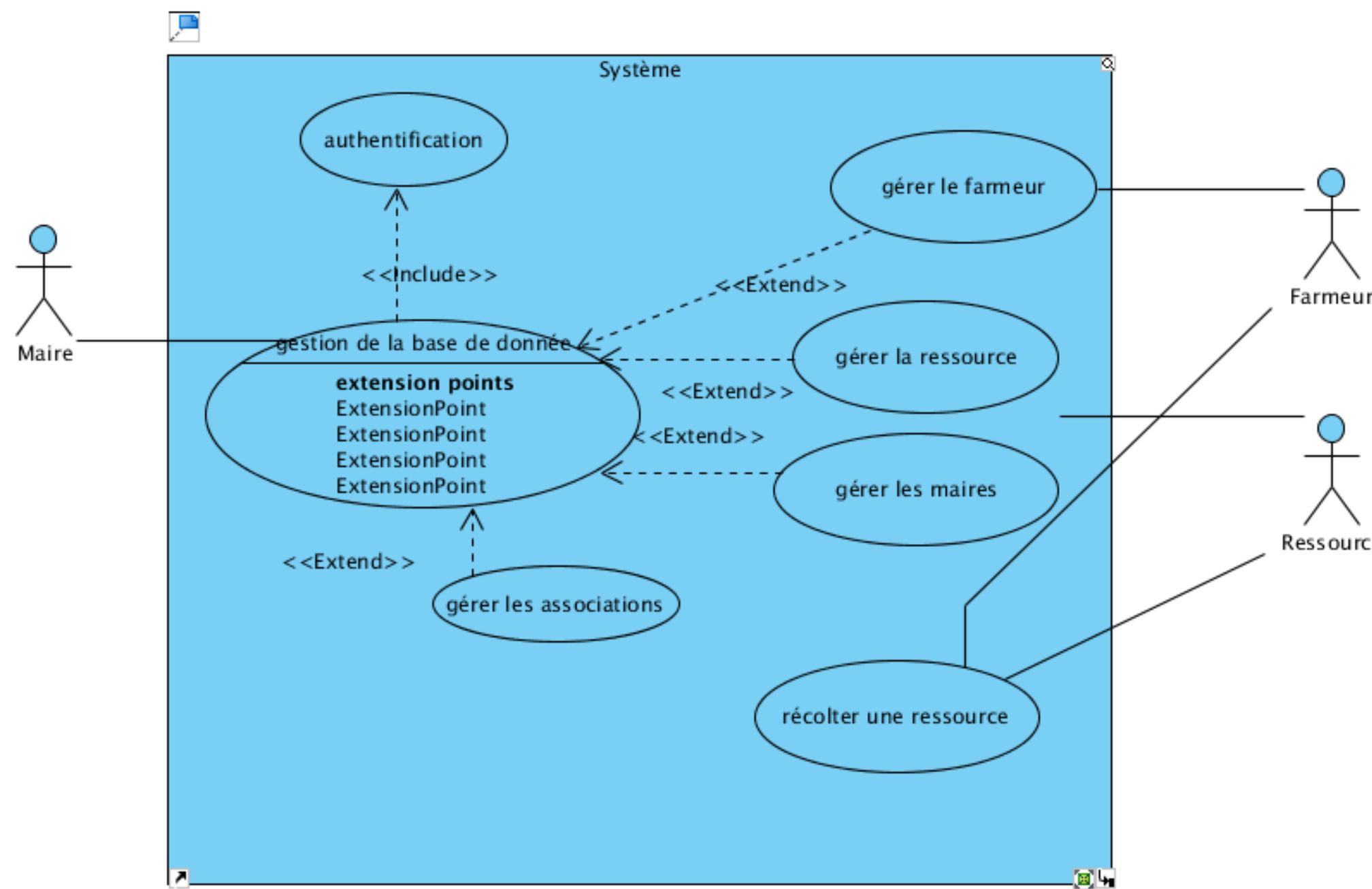


DIAGRAMME UML

Entité

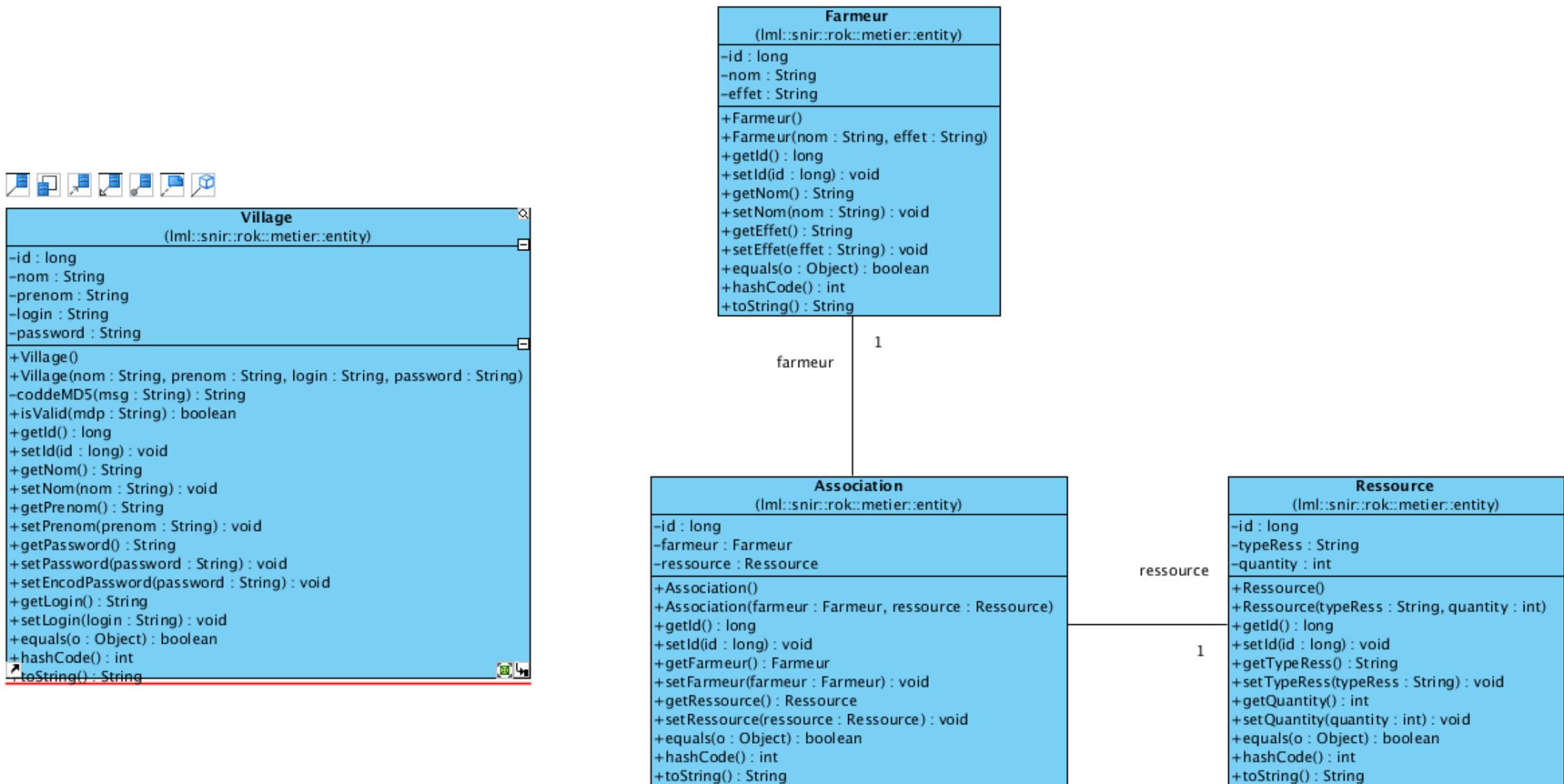
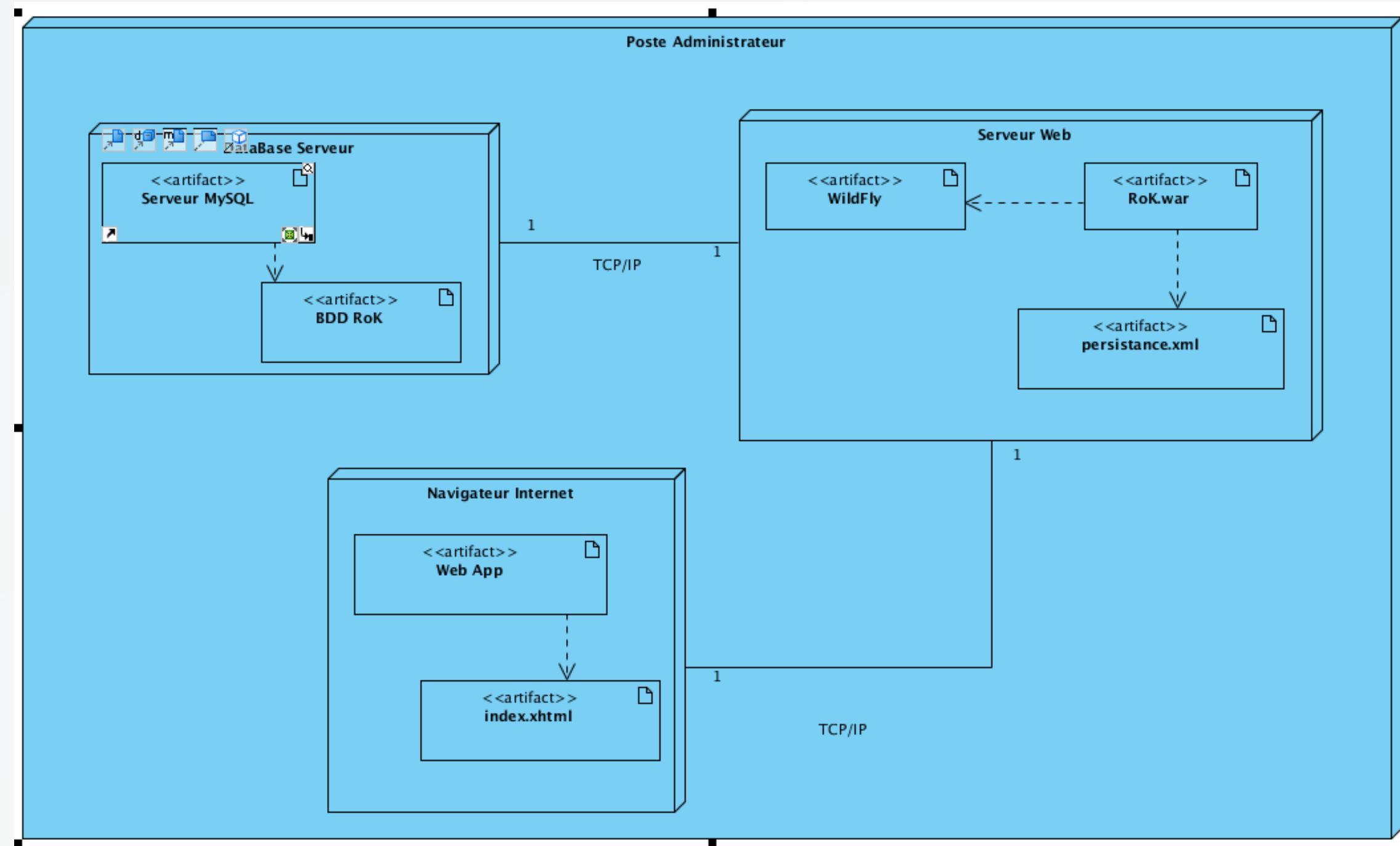


DIAGRAMME UML

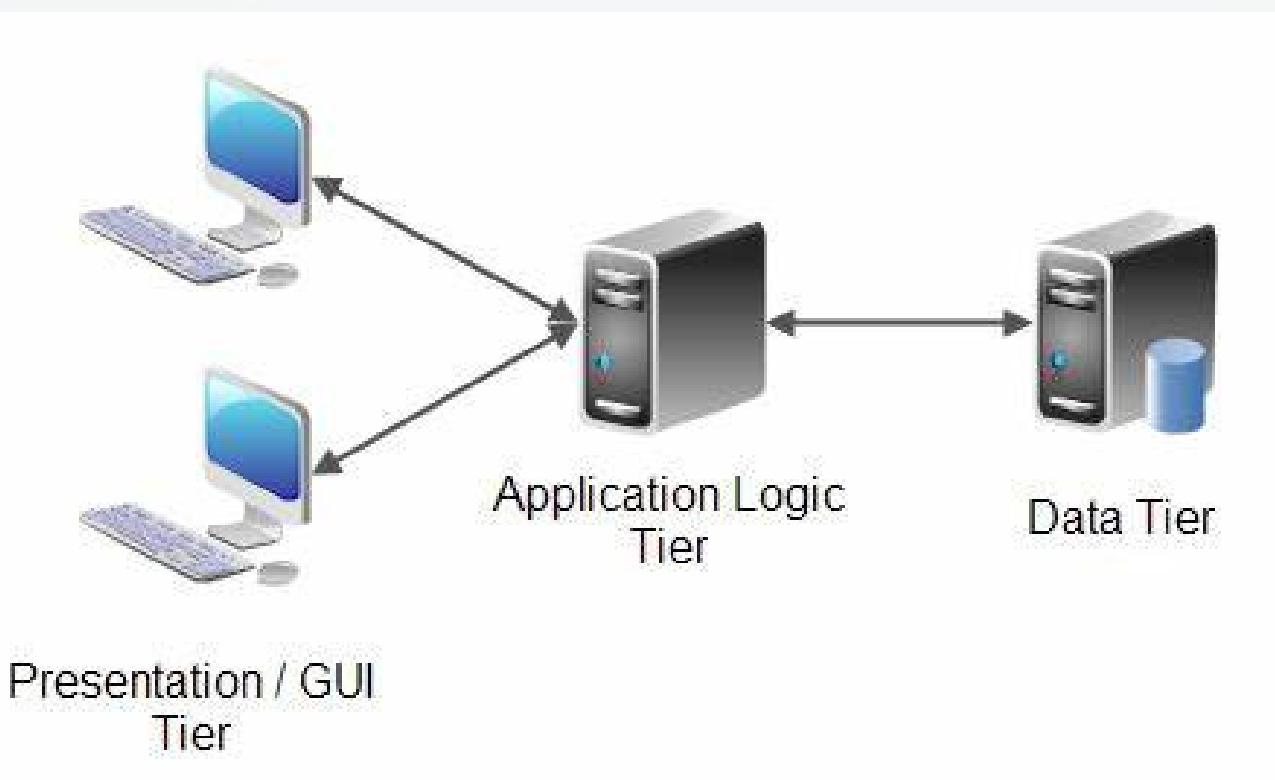
Déploiement



LES CLASSES

CLASSES

- VILLAGE
- FARMEUR
- RESSOURCE
- ASSOCIATION



Architecture en n-tiers

-	lml.smir.rok.client.bean
-	FarmeurFactory.java
-	FarmeurServiceBean.java
-	IndexBean.java
-	LoginBean.java
-	MaireServiceBean.java
-	NavigatorBean.java
-	RessourceServiceBean.java
-	lml.smir.rok.converter
-	FarmeurConverter.java
-	RessourceConverter.java
-	lml.smir.rok.metier.entity
-	Association.java
-	Farmeur.java
-	Ressource.java
-	Village.java
-	lml.smir.rok.metier.transactionn
-	AssociationService.java
-	AssociationServiceImpl.java
-	FarmeurService.java
-	FarmeurServiceImpl.java
-	RessourceService.java
-	RessourceServiceImpl.java
-	VillageService.java
-	VillageServiceImpl.java
-	lml.smir.rok.metierfactory
-	MetierFactory.java
-	lml.smir.rok.physique.data

LA CLASSE ASSOCIATION

```
@OneToOne  
private Farmer farmer;  
  
@OneToOne  
private Ressource ressource;  
  
public Association() {  
}  
  
public Association(Farmer farmer, Ressource ressource) {  
    this.farmer = farmer;  
    this.ressource = ressource;  
}  
  
public long getId() {  
    return id;  
}  
  
public void setId(long id) {  
    this.id = id;  
}  
  
public Farmer getFarmer() {  
    return farmer;  
}  
  
public void setFarmer(Farmer farmer) {  
    this.farmer = farmer;  
}  
  
public Ressource getRessource() {  
    return ressource;  
}  
  
public void setRessource(Ressource ressource) {  
    this.ressource = ressource;  
}  
  
@Override  
public boolean equals(Object o) {  
    if (o == null) {  
        return false;  
    }  
    if (!(o instanceof Farmer)) {  
        throw new ClassCastException();  
    }  
    return (o.hashCode() == this.hashCode());  
}
```

LA COUCHE PHYSIQUE

```
/*
public interface AssociationDataService extends CrudService<Association> {
    public Association getByFarmeur(Farmeur ressource) throws Exception;
    public Association getByRessource(Ressource ressource) throws Exception;
}

public AssociationDataServiceJPAImpl(){
    super(PU:"lml.snir.ROKPU");
}

@Override
public Association getByFarmeur(Farmeur farmeur) throws Exception{
Association association = null;
try {
    this.open();
    Query query = em.createQuery(qlString:"SELECT association FROM Association association WHERE association.farmeur = :ffarmeuroAssociation");
    query.setParameter(name: "ffarmeuroAssociation", value: farmeur);
    association = (Association) query.getSingleResult();
} catch (NoResultException ex) {
    return null;
} finally {
    this.close();
}
return association;
}

@Override
public Association getByRessource(Ressource ressource) throws Exception {
Association association = null;
try {
    this.open();
    Query query = em.createQuery(qlString:"SELECT association FROM Association association WHERE association.ressource = :fressourceAssociation");
    query.setParameter(name: "fressourceAssociation", value: ressource);
    association = (Association) query.getSingleResult();
} catch (NoResultException ex) {
    return null;
}
```

Le bean et son code XHTML

```

<p:panel>
    <p:dataTable id="datat" var="assos" value="#{indexBean.associations}" allowUnsorting="true" sortMode="single">
        <p:column headerText="Nom" width="20" sortBy="#{assos.farmer.nom}" >
            <h:outputText value="#{assos.farmer.nom}" />
        </p:column>
        <p:column headerText="Effet" width="100" sortBy="#{assos.farmer.effet}" >
            <h:outputText value="#{assos.farmer.effet}" />
        </p:column>
        <p:column headerText="Ressource" width="100" sortBy="#{assos.ressource.typeRess}" >
            <h:outputText value="#{assos.ressource.typeRess}" />
        </p:column>
        <p:column headerText="Quantité" width="100" sortBy="#{assos.ressource.quantity}" >
            <h:outputText value="#{assos.ressource.quantity}" />
        </p:column>
        <p:column headerText="Éditer / Supprimer" width="100" >
            <p:commandButton id="dynaButton" value="Action" type="button" icon="pi pi-home" />
        </p:column>
    </p:dataTable>
    <p:tieredMenu overlay="true" trigger="dynaButton" my="left top" at="left bottom" style="width: 200px">
        <p:submenu label="Modifier">
            <p:menuItem value="#{assos.toString()}" actionListener="#{indexBean.modifierAssos(assos)}" action="#{navigatorBean.modifierAssos()}" ajax="false" />
        </p:submenu>
        <p:submenu label="Supprimer" >
            <p:menuItem value="#{assos.toString()}" actionListener="#{indexBean.supprimerAssos(assos)}" action="#{navigatorBean.ensembleService()}" ajax="false" >
                <p:confirm header="Confirmation" message="Voulez-vous vraiment supprimer cette association ?" icon="pi pi-info-circle"/>
            </p:menuItem>
        </p:submenu>
    </p:tieredMenu>
    </p:column>
</p:panel>

<p:dialog modal="true" header="Ajouter une association" widgetVar="dlgAdd" >
    <ui:insert name="ajouter" >
        <ui:include id="ajouterAssos" src="/Acceuil/ajouterAssos.xhtml" />
    </ui:insert>
</p:dialog>

<p:confirmDialog global="true" showEffect="fade" hideEffect="fade" responsive="true" width="350">
    <p:commandButton value="Non" type="button" styleClass="ui-confirmdialog-no ui-button-flat"/>
    <p:commandButton value="Oui" type="button" styleClass="ui-confirmdialog-yes ui-button-danger"/>
</p:confirmDialog>

<p:panel>
    <p:commandButton value="Ajouter" onclick="PF('dlgAdd').show()" />
</p:panel>

```

Nom ↑	Effet ↑	Ressource ↑	Quantité ↑	
Jeanne d'arc	speedwalking	pierre		Action
Jeanne d'arc	speedwalking	bois		Action
Constance	capacity	pierre		Action
Constance	capacity	pierre		Action
Caius Marius	speedmining	pierre		Action
Caius Marius	speedmining	blé		Action
Caius Marius	speedmining	blé	789	Action
Constance	capacity	bois	857	Action

Ajouter une association

Associez

Farmer : Sélectionnez un farmer

Ressource

1 : Jeanne d'arc
2 : Constance
3 : Caius Marius

Ajouter

```

<html xmlns:h="http://java.sun.com/jsf/html">
    <head>
        <title>TODO supply a title</title>
        <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
    </head>
    <body>
        <ui:composition>
            <h2>Associez</h2>

            <p:panelGrid columns="2" layout="tabular" >
                <p:outputLabel for="ajouter" value="Farmer : " />
                <p:selectOneMenu id="ajouter" value="#{indexBean.selectedFarmer}" converter="#{ressourceConverter}" >
                    <f:selectItem itemValue="" itemLabel="Sélectionnez un farmer" />
                    <f:selectItems value="#{indexBean.farmers}" itemLabel="#{farmers}" />
                    <p:ajax event="itemSelect" update="@this" />
                </p:selectOneMenu>

                <p:outputLabel for="ajouter2" value="Ressource " />
                <p:selectOneMenu id="ajouter2" value="#{indexBean.selectedRessource}" >
                    <f:selectItem itemValue="" itemLabel="Sélectionnez une ressource" />
                    <f:selectItems value="#{indexBean.ressources}" itemLabel="#{selectedRessource}" />
                    <p:ajax event="itemSelect" update="@this" />
                </p:selectOneMenu>

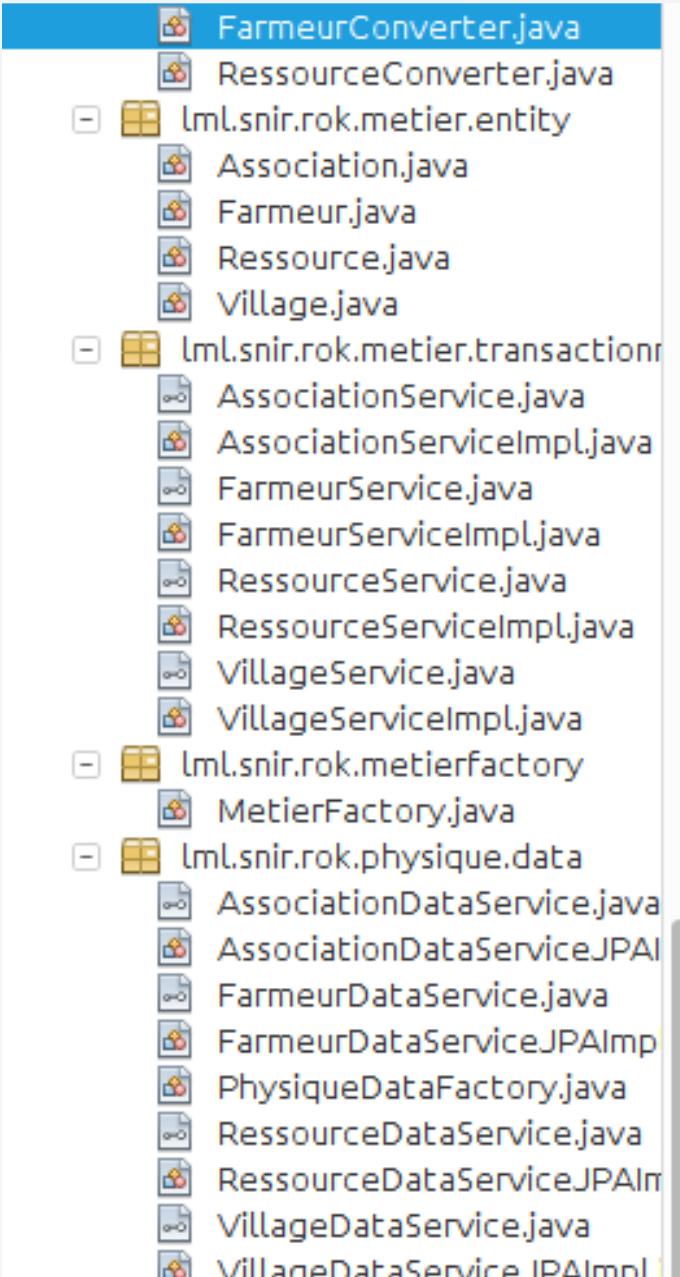
                <br />
                <p:commandButton value="Ajouter" actionListener="#{indexBean.ajouterAssos()}" action="#{navigatorBean.ensembleService()}" ajax="false" />
            </p:panelGrid>
        </ui:composition>

```



MENTIONS HONORABLES

LES CONVERTERS



```
@FacesConverter(forClass = Farmeur.class)
public class FarmeurConverter implements Converter {

    @Override
    public Object getAsObject(FacesContext fc, UIComponent uic, String string) {
        Farmeur farm = null;

        if (string == null || string.length() == 0) {
            return null;
        }

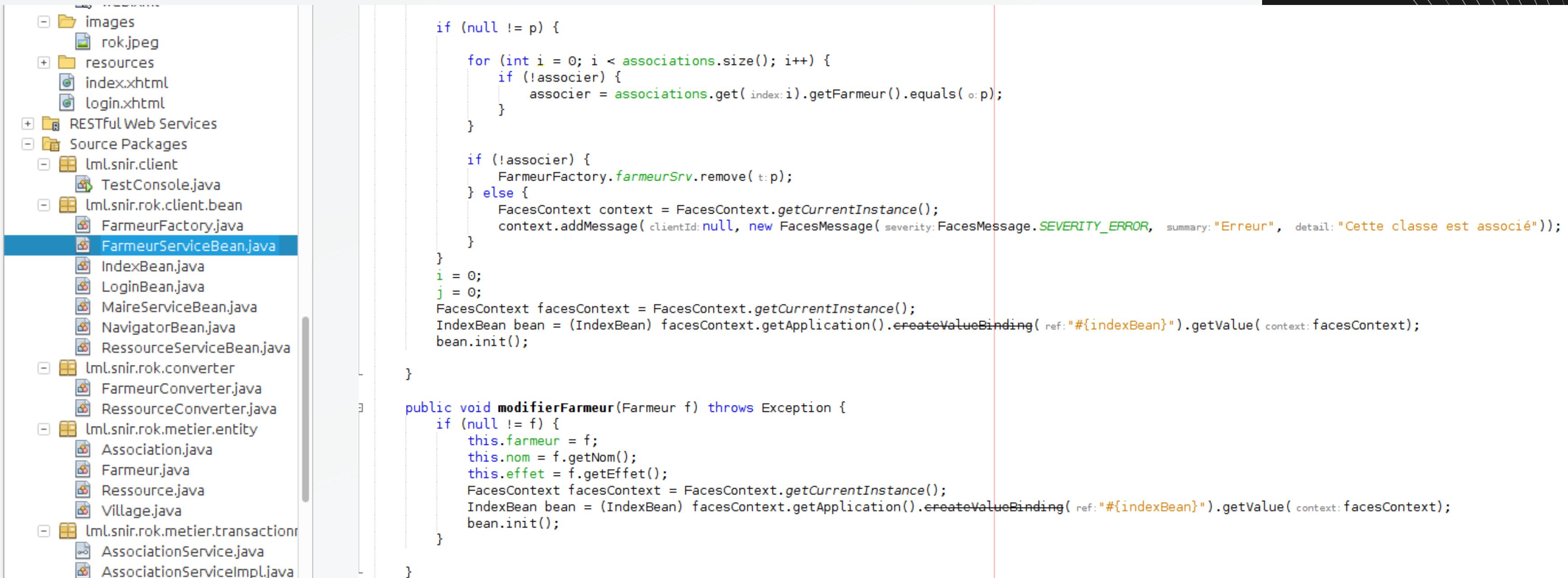
        try {
            farm = PhysiqueDataFactory.getFarmeurDataService().getById((long) Integer.parseInt(string.split(" : ")[0]));
            System.out.println(x: farm);
        } catch (Exception e) {
            Logger.getLogger(FarmeurConverter.class.getName()).log(Level.SEVERE, null, e);
        }
    }

    return farm;
}

@Override
public String getAsString(FacesContext fc, UIComponent uic, Object o) {
    Farmeur farm = (Farmeur) o;

    return farm.toString();
}
}
```

FACECONTEXT

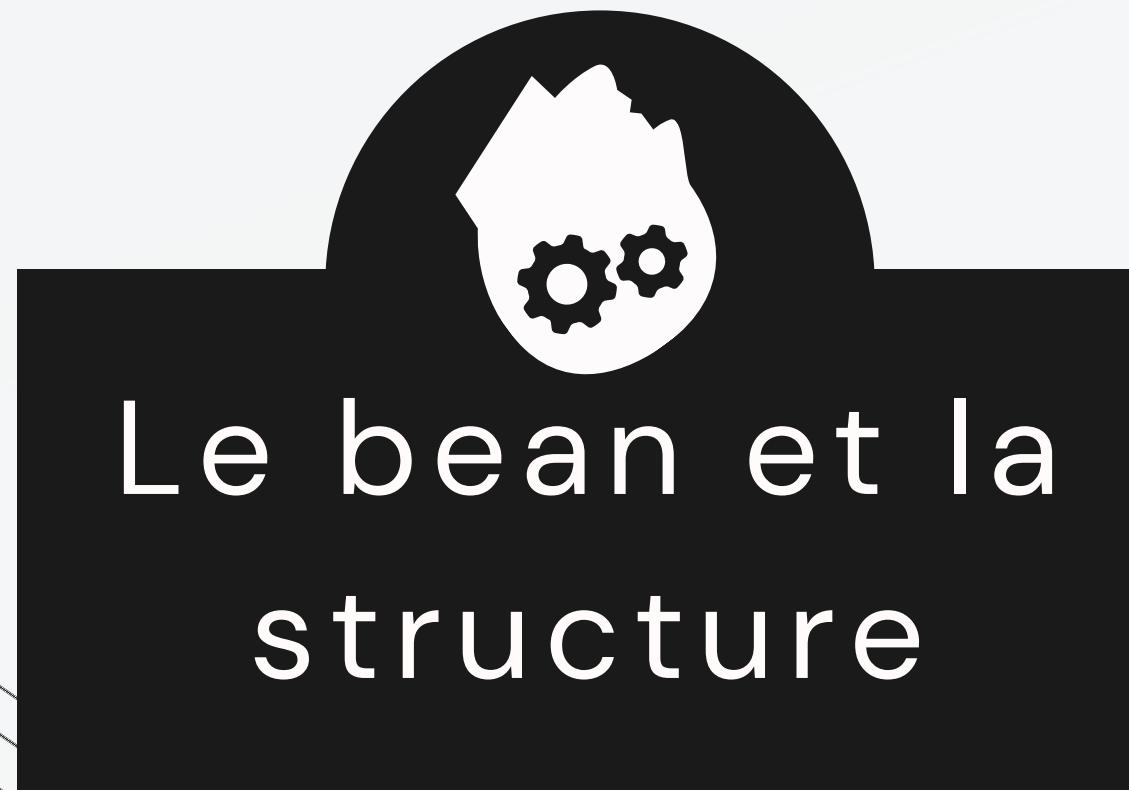


The screenshot shows a Java development environment with a file tree on the left and a code editor on the right.

File Tree:

- images
- resources
- index.xhtml
- login.xhtml
- RESTful Web Services
- Source Packages
 - lml.snir.client
 - TestConsole.java
 - lml.rok.client.bean
 - FarmeurFactory.java
 - FarmeurServiceBean.java
 - IndexBean.java
 - LoginBean.java
 - MaireServiceBean.java
 - NavigatorBean.java
 - RessourceServiceBean.java
- lml.rok.converter
 - FarmeurConverter.java
 - RessourceConverter.java
- lml.rok.metier.entity
 - Association.java
 - Farmeur.java
 - Ressource.java
 - Village.java
- lml.rok.metier.transaction
 - AssociationService.java
 - AssociationServiceImpl.java

PROBLÈME RENCONTRÉ



**MERCI DE
M'AVOIR
ÉCOUTÉ**

Avez vous des question ?

