



Site de E-Commerce avec Spring Framework

Youssfi Mohamed

ENSET

Université Hassan II Mohammedia

med@yousfsi.net

Problème

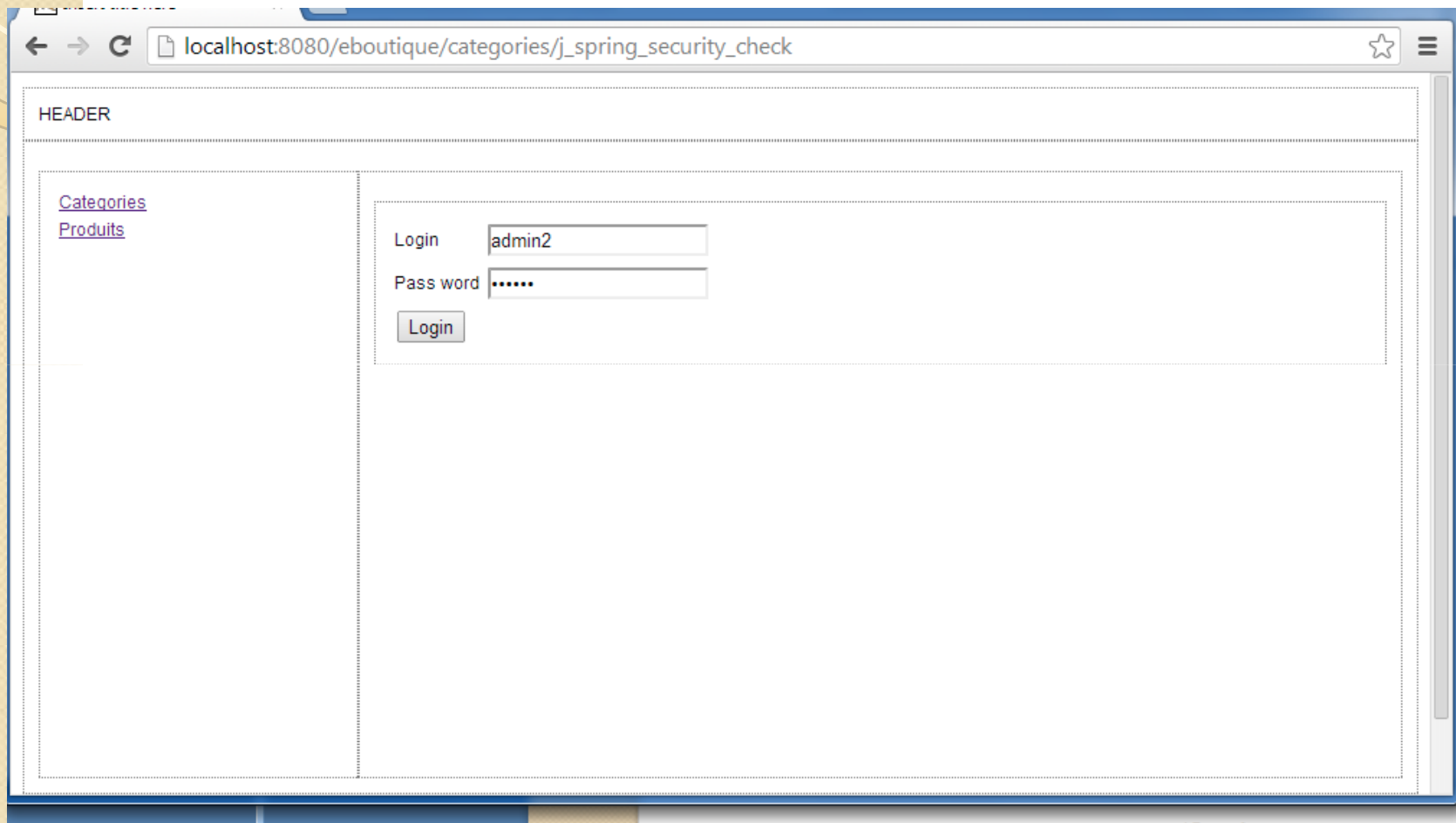
- On souhaite créer un site web de commerce électronique qui se compose de deux parties :
 - Une partie back office qui nécessite une authentification et qui permet de gérer les produits et les catégories
 - Nous définissons deux rôles pour cette partie:
 - `ROLE_ADMIN_CAT` : l'utilisateur ayant ce rôle a la possibilité de gérer les catégories et les produits (Ajout, suppression, Edition, Modification et consultation) ainsi que les droits d'accès
 - `ROLE_ADMIN_PROD` : l'utilisateur ayant ce rôle a la possibilité de gérer uniquement les produits (Ajout, suppression, Edition, Modification et consultation)
 - Une partie front office qui représente la boutique virtuelle qui ne nécessite pas d'authentification. Dans cette partie l'utilisateur a la possibilité de :
 - Consulter toutes les catégories
 - Consulter les produits d'une catégorie
 - Consulter les produits sélectionnés
 - Chercher des produits par mot clé
 - Ajouter un produit avec une quantité au panier
 - Supprimer un produit du panier
 - Enregistrer le client et la commande des produits de son panier.

Gestion des catégories

The screenshot shows a web browser window with the address bar displaying `localhost:8080/eboutique/categories/index`. The page layout includes a header section and a main content area. On the left side of the main content area, there is a sidebar with two links: [Categories](#) and [Produits](#). The main content area contains a login form with the following elements:

- A label "Login" followed by a text input field.
- A label "Pass word" followed by a text input field.
- A "Login" button below the input fields.

Gestion des catégories



The screenshot shows a web browser window with the address bar displaying `localhost:8080/eboutique/categories/j_spring_security_check`. The page content is enclosed in a dotted border and includes a header section labeled "HEADER". Below the header, there are two links: [Categories](#) and [Produits](#). To the right of these links is a login form with the following fields and controls:

- Login:
- Pass word:
- Login:

Gestion des catégories

Insert title here x

localhost:8080/eboutique/categories/index

HEADER



[Categories](#)
[Produits](#)

ID Catégorie:

Nom Catégorie

Description

Photo Aucun fichier choisi

ID	NOM CAT	Description	Photo		
1	Ordinateurs	Ordinateurs		Supprimer	Edit
2	Imprimantes	Imprimantes		Supprimer	Edit

Gestion des produits

HEADER

[Categories](#)
[Produits](#)

[Logout](#)

ID Produit

Categorie

Désignation



Description

Prix


Sélectionné ☐


Quantité

Photo Aucun fichier choisi

ID	Designation	Prix	Sélectionnée	Quatité	Photo		
1	HP5600	8000.0	true	12		Supp	Edit
2	Compaq 76800	9000.0	true	80		Supp	Edit

Partie Front Office






Nombre de produits : 5
Total : 44000.0


[Index](#)

[Ordinateurs](#)
[Importantes](#)
[Téléphones](#)


ID	Désignation	Prix	Quantité	Montant
1	HP5600	8000.0	1	8000.0
2	Compaq 76800	9000.0	4	36000.0
Total				44000.0



Désignation : HP5600
Prix : 8000.0
Stock : 12
HP90876



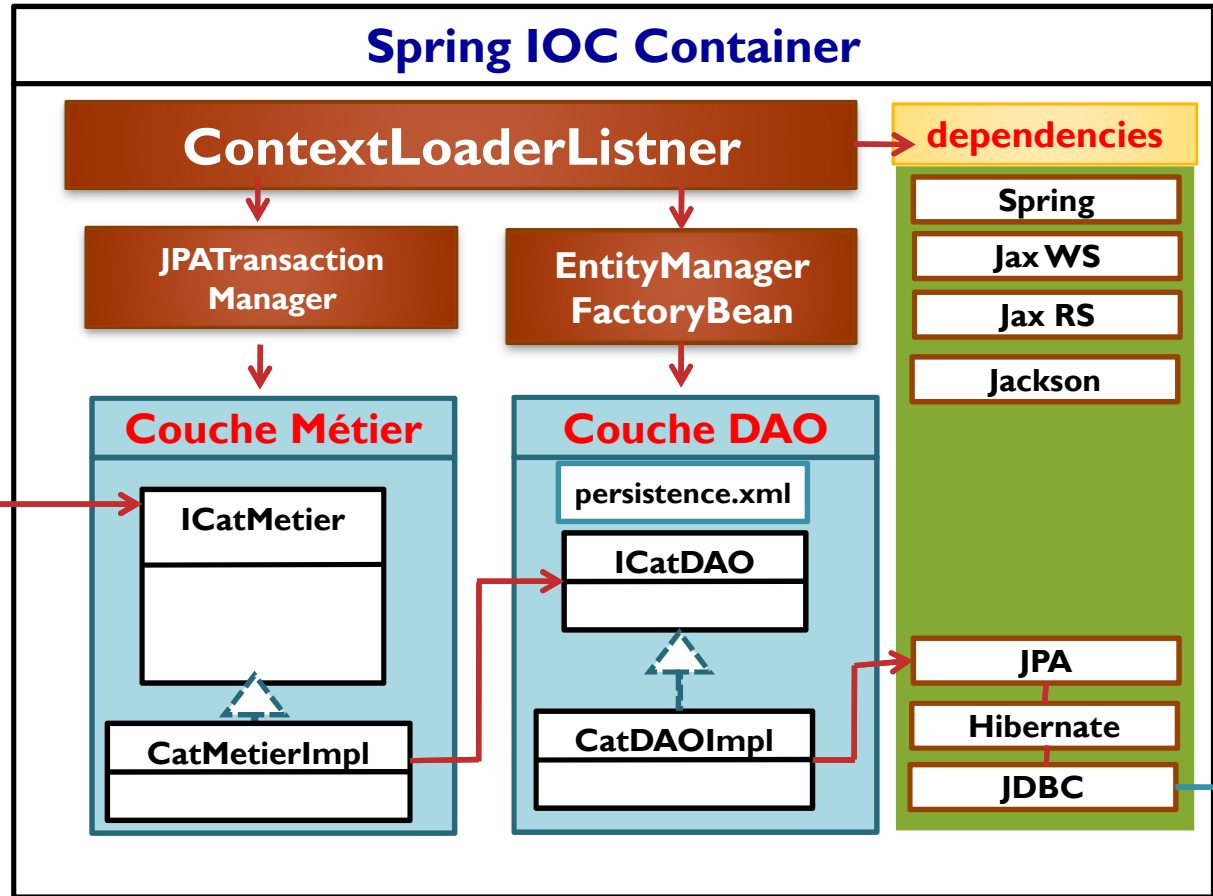
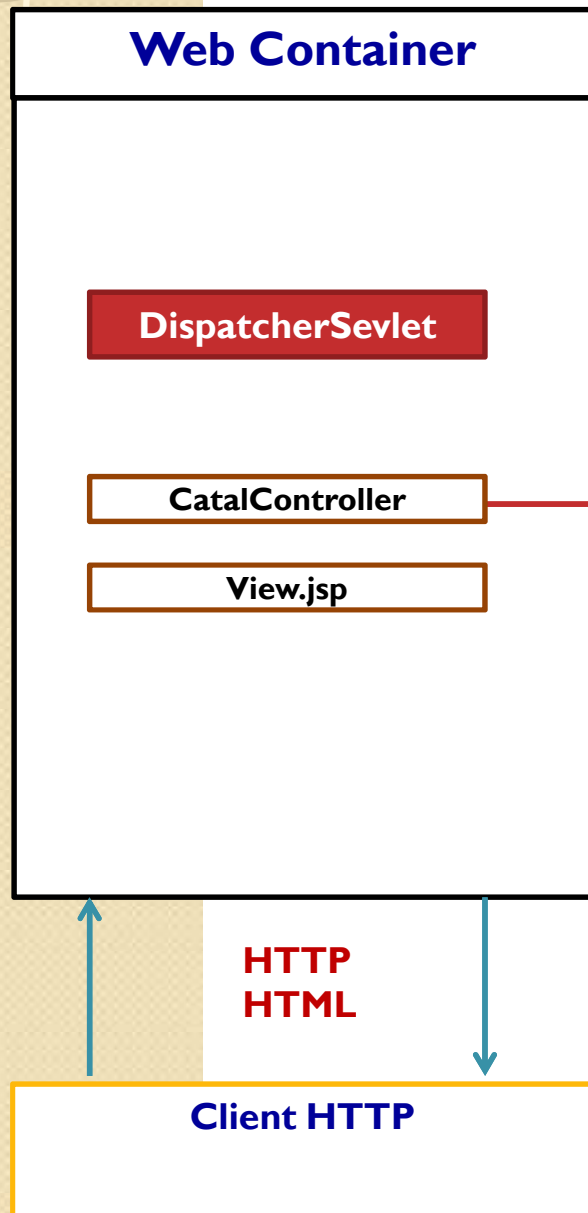
Désignation : Compaq 76800
Prix : 9000.0
Stock : 80



Désignation : HP Jet 8700
Prix : 6500.0
Stock : 3

FOOTER

Architecture



Use Case

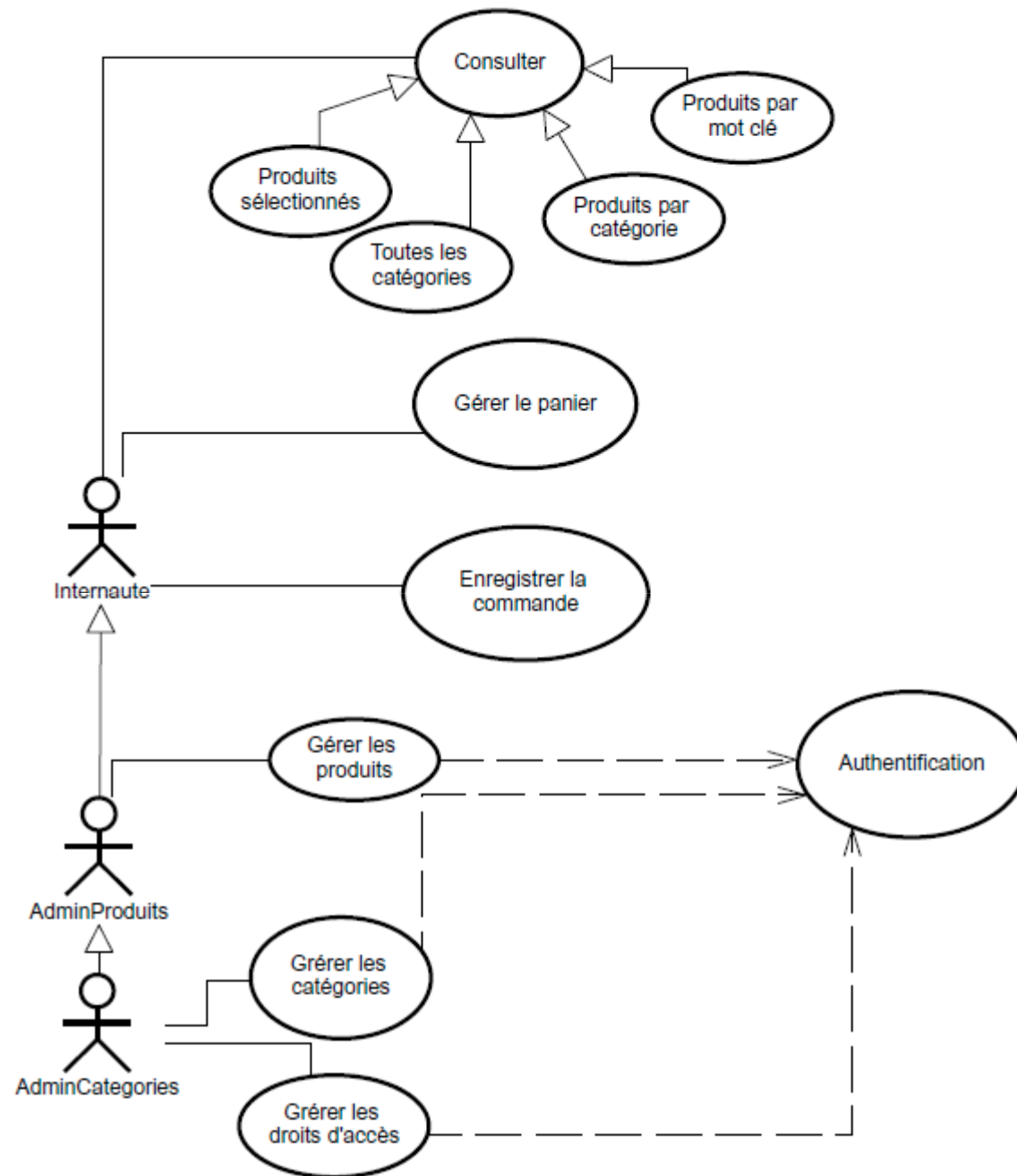


Diagramme de classes des entités

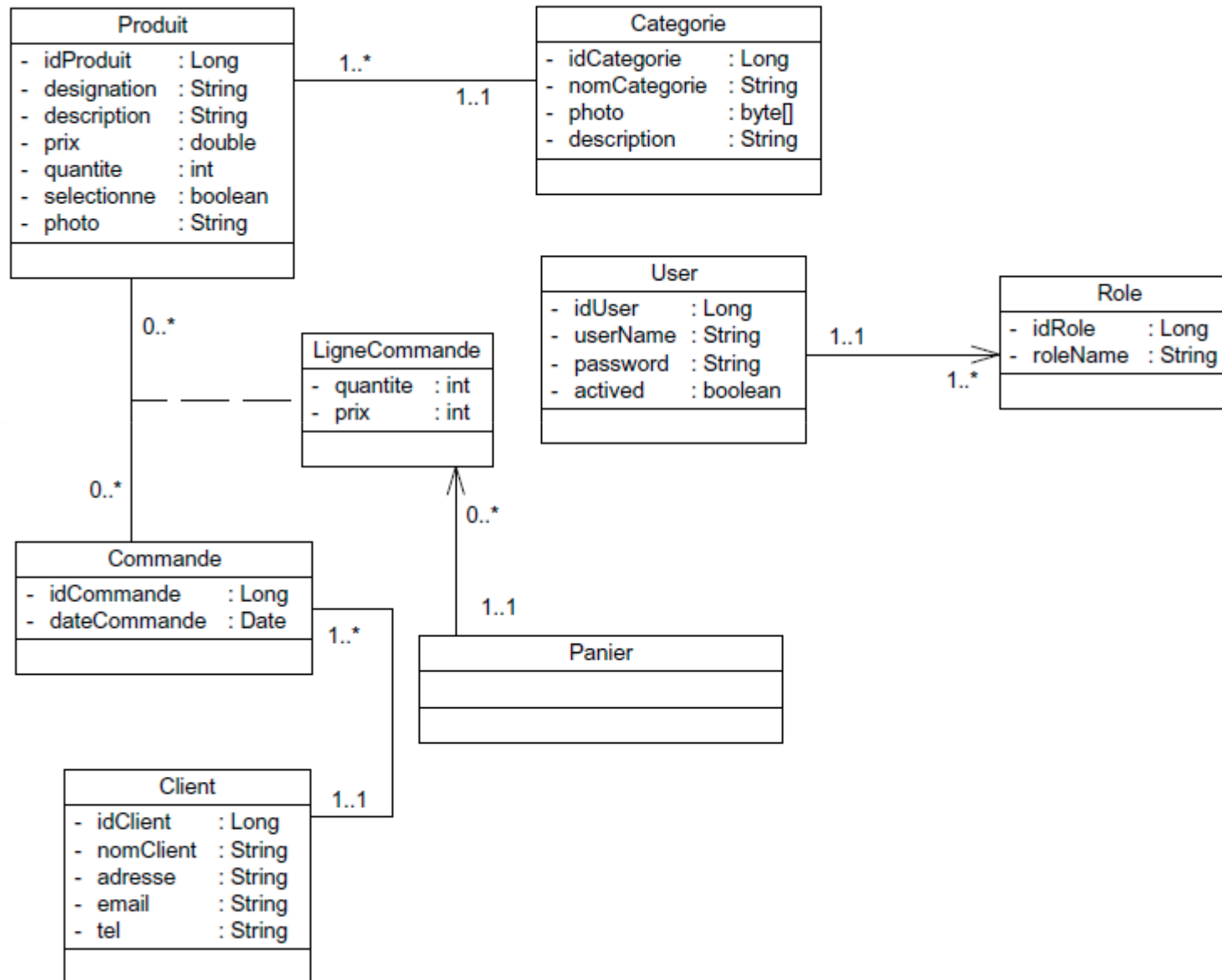
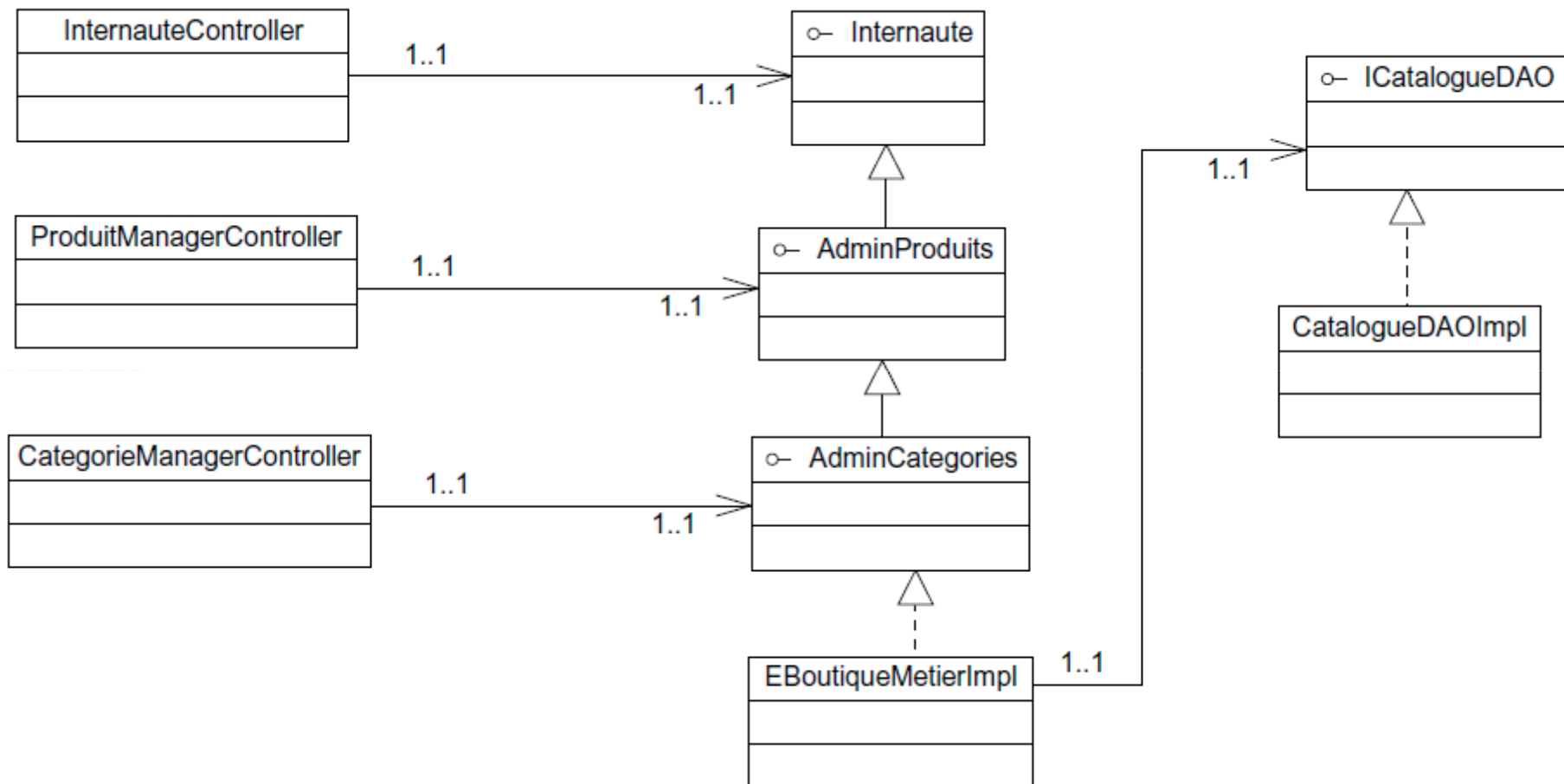


Diagramme de classes des traitements





Création des entités

Maven Dependencies

```
<!-- Hibernate-->
```

```
<dependency>
```

```
    <groupId>org.hibernate</groupId>
```

```
    <artifactId>hibernate-entitymanager</artifactId>
```

```
    <version>3.6.0.Final</version>
```

```
</dependency>
```

```
<dependency>
```

```
    <groupId>org.hibernate</groupId>
```

```
    <artifactId>hibernate-validator</artifactId>
```

```
    <version>4.1.0.Final</version>
```

```
</dependency>
```

```
<!-- MySQL Connector-->
```

```
<dependency>
```

```
    <groupId>mysql</groupId>
```

```
    <artifactId>mysql-connector-java</artifactId>
```

```
    <version>5.1.6</version>
```

```
</dependency>
```

Categorie.java

```
package net.youssfi.eboutique.entities;
import java.io.Serializable;import java.util.*;
import javax.persistence.*;import javax.validation.constraints.Size;
import org.hibernate.validator.constraints.NotEmpty;
@Entity
public class Categorie implements Serializable {
    @Id
    @GeneratedValue
    private Long idCategorie;
    @NotEmpty
    @Size(min=4,max=20)
    private String nomCategorie; private String description;
    private String nomPhoto;
    @Lob
    private byte[] photo;
    @OneToMany(mappedBy="categorie")
    private Collection<Produit> produits=new ArrayList<Produit>();
    // Getters et Setters
    // Constructeurs sans paramètre et avec paramètres
}
```

Produit.java

```
package net.youssfi.eboutique.entities;
import java.io.Serializable; import javax.persistence.*;
import org.hibernate.validator.constraints.NotEmpty;
@Entity
public class Produit implements Serializable {
    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private Long idProduit;
    @NotEmpty
    private String designation; private String description;
    private double prix; private String photo;
    private int quantite; private boolean selectionne;
    @ManyToOne
    @JoinColumn(name="ID_CAT")
    private Categorie categorie;
    // Getters et Setters
    // Cosntructeur sans paramètre
    // Cosntructeur avec paramètres
}
```


Client.java

```
package net.youssfi.eboutique.entities;
import java.io.Serializable;
import java.util.Collection;
import javax.persistence.*;
@Entity
public class Client implements Serializable {
    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    private Long idClient;
    private String nomClient;
    private String adresse;
    private String email;
    private String tel;
    @OneToMany(mappedBy="client")
    private Collection<Commande> commandes;
    // Getters et Setters
    // Cosntructeur sans paramètre
    // Cosntructeur avec paramètres
}
```

Commande.java

```
package net.youssfi.eboutique.entities;
import java.io.Serializable;
import java.util.*;
import javax.persistence.*;
@Entity
public class Commande implements Serializable {
    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    private Long idCommande;
    private Date dateCommande;
    @ManyToOne
    @JoinColumn(name="idClient")
    private Client client;
    @OneToMany
    @JoinColumn(name="idCommande")
    private Collection<LigneCommande> ligneCommandes;
    // Getters et Setters
    // Cosntructeur sans paramètre
    // Cosntructeur avec paramètres
}
```

LigneCommande.java

```
package net.youssfi.eboutique.entities;
import java.io.Serializable;
import javax.persistence.*;
@Entity
public class LigneCommande implements Serializable {
    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    private Long id;
    @ManyToOne
    @JoinColumn(name="idproduit")
    private Produit produit;
    private int quantite;
    private double prix;
    // Getters et Setters
    // Cosntructeur sans paramètre
    // Cosntructeur avec paramètres
}
```

Panier.java

```
package net.youssfi.eboutique.entities;
import java.io.Serializable;
import java.util.*;
public class Panier implements Serializable {
    private Map<Long, LigneCommande> items=new HashMap<Long, LigneCommande>();
    public void addItem(Produit p, int quantite){
        LigneCommande lc=items.get(p.getIdProduit());
        if(lc==null){
            LigneCommande art=new LigneCommande();
            art.setProduit(p);
            art.setQuantite(quantite);
            art.setPrix(p.getPrix());
            items.put(p.getIdProduit(), art);
        }
        else{
            lc.setQuantite(lc.getQuantite()+quantite);
        }
    }
}
```

Panier.java

```
public Collection<LigneCommande> getItems(){
    return items.values();
}

public int getSize(){
    return items.size();
}

public double getTotal(){
    double total=0;
    for(LigneCommande lc:items.values()){
        total+=lc.getPrix()*lc.getQuantite();
    }
    return total;
}

public void deleteItem(Long idproduit){
    items.remove(idproduit);
}
}
```



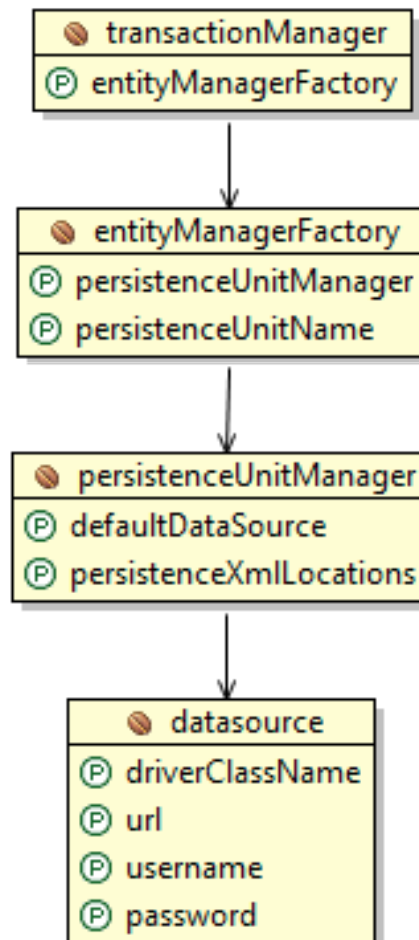
Tester les entités

/EBoutiqueV2/src/main/resources/META-INF/persistence.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.0" xmlns="http://java.sun.com/xml/ns/persistence"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
    http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd ">
  <persistence-unit name="UP_EBOUTIQUE" transaction-type="RESOURCE_LOCAL">
    <provider>org.hibernate.ejb.HibernatePersistence</provider>
    <properties>
      <property name="hibernate.show_sql" value="true"/>
      <property name="hibernate.hbm2ddl.auto" value="update"/>
      <property name="hibernate.dialect"
        value="org.hibernate.dialect.MySQLDialect"/>
    </properties>
  </persistence-unit>
</persistence>
```


Configuration JPA avec Spring IOC

Spring Beans



Maven Properties

```
<properties>  
<java-version>1.6</java-version>  
<org.springframework-  
  version>3.2.2.RELEASE</org.springframework-version>  
<org.aspectj-version>1.6.10</org.aspectj-version>  
<org.slf4j-version>1.6.6</org.slf4j-version>  
</properties>
```

Maven Dependencies

```
<!-- Spring -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-context</artifactId>
  <version>${org.springframework-version}</version>
  <exclusions>
    <!-- Exclude Commons Logging in favor of SLF4j -->
    <exclusion>
      <groupId>commons-logging</groupId>
      <artifactId>commons-logging</artifactId>
    </exclusion>
  </exclusions>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-webmvc</artifactId>
  <version>${org.springframework-version}</version>
</dependency>
```

Maven Dependencies

```
<!-- Spring -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-tx</artifactId>
  <version>>${org.springframework-version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-orm</artifactId>
  <version>>${org.springframework-version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-beans</artifactId>
  <version>>${org.springframework-version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-core</artifactId>
  <version>>${org.springframework-version}</version>
</dependency>
```

Configuration JPA avec Spring IOC :

/EBoutiqueV2/src/main/resources/applicationContext.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:context="http://www.springframework.org/schema/context"
xmlns:tx="http://www.springframework.org/schema/tx"
xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.2.xsd
http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-3.2.xsd">

<bean id="datasource"
    class="org.springframework.jdbc.datasource.DriverManagerDataSource">
    <property name="driverClassName" value="com.mysql.jdbc.Driver"></property>
    <property name="url" value="jdbc:mysql://localhost:3306/eboutique"></property>
    <property name="username" value="root"></property>
    <property name="password" value=""></property>
</bean>
```

Configuration JPA avec Spring IOC :

/EBoutiqueV2/src/main/resources/applicationContext.xml

```
<bean id="persistenceUnitManager"
    class="org.springframework.orm.jpa.persistenceunit.DefaultPersistenceUnitManager">
    <property name="defaultDataSource" ref="datasource"></property>
    <property name="persistenceXmlLocations">
        <list>
            <value>classpath*:META-INF/persistence.xml</value>
        </list>
    </property>
</bean>

<bean id="entityManagerFactory"
    class="org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean">
    <property name="persistenceUnitManager" ref="persistenceUnitManager"></property>
    <property name="persistenceUnitName" value="UP_EBOUTIQUE"></property>
</bean>

    <bean id="transactionManager"
        class="org.springframework.orm.jpa.JpaTransactionManager">
            <property name="entityManagerFactory" ref="entityManagerFactory"></property>
        </bean>
    <tx:annotation-driven transaction-manager="transactionManager"/>
    <context:annotation-config/>
</beans>
```




JUnit Test : src/test/java/TestDao.java

```
package net.youssfi.eboutique;
import static org.junit.Assert.*;
import org.junit.*;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class TestDAO {
    @Before
    public void setUp() throws Exception {
    }
    @Test
    public void test() {
        try {
            ClassPathXmlApplicationContext app=
new ClassPathXmlApplicationContext(new String[]{"applicationContext.xml"});
            assertTrue(true);
        } catch (Exception e) {
            assertTrue(e.getMessage(),false);
        }
    }
}
```


Tables générées



The screenshot shows a database management interface with a tabbed menu at the top containing 'Structure', 'SQL', and 'Rechercher'. Below the menu is a table listing generated tables. Each row includes a checkbox, the table name, an 'Afficher' (View) button, and a 'Structure' button. The table 'lignecommande' is currently selected. At the bottom, a summary row indicates '7 tables' and a 'Somme' (Total) button.

Table	Action
<input type="checkbox"/> categorie	<input type="checkbox"/> Afficher  Structure
<input type="checkbox"/> client	<input type="checkbox"/> Afficher  Structure
<input type="checkbox"/> commande	<input type="checkbox"/> Afficher  Structure
<input checked="" type="checkbox"/> lignecommande	<input type="checkbox"/> Afficher  Structure
<input type="checkbox"/> produit	<input type="checkbox"/> Afficher  Structure
<input type="checkbox"/> roles	<input type="checkbox"/> Afficher  Structure
<input type="checkbox"/> users	<input type="checkbox"/> Afficher  Structure
7 tables	Somme



COUCHE DAO

Inteface IBoutiqueDao

```
package net.youssfi.eboutique.dao;
import java.util.List;
import net.youssfi.eboutique.entities.*;
public interface IBoutiqueDao {
    public Long ajouterCategorie(Categorie c);
    public List<Categorie> listCategories();
    public Categorie getCategorie(Long idCat);
    public void supprimerCategorie(Long idcat);
    public void modifierCategorie(Categorie c);

    public Long ajouterProduit(Produit p, Long idCat);
    public List<Produit> listproduits();
    public List<Produit> produitsParMotCle(String mc);
    public List<Produit> produitsParCategorie(Long idCat);
    public List<Produit> produitsSelectionnes();
    public Produit getProduit(Long idP);
    public void supprimerProduit(Long idP);
    public void modifierProduit(Produit p);

    public void ajouterUser(User u);
    public void attribuerRole(Role r,Long userID);
    public Commande enregistrerCommande(Panier p,Client c);
}
```

BoutiqueDaoImpl

```
package net.youssfi.eboutique.dao;

import java.util.List;
import javax.persistence.*;
import net.youssfi.eboutique.entities.*;

public class BoutiqueDaoImpl implements IBoutiqueDao {
    @PersistenceContext
    private EntityManager em;

    @Override
    public Long ajouterCategorie(Categorie c) {
        em.persist(c);
        return c.getIdCategorie();
    }

    @Override
    public List<Categorie> listCategories() {
        Query req=em.createQuery("select c from Categorie c");
        return req.getResultList();
    }
}
```

BoutiqueDaolImpl

```
@Override
public Categorie getCategorie(Long idCat) {
    return em.find(Categorie.class, idCat);
}

@Override
public void supprimerCategorie(Long idcat) {
    Categorie c=em.find(Categorie.class, idcat);
    em.remove(c);
}

@Override
public void modifierCategorie(Categorie c) {
    em.merge(c);
}

@Override
public Long ajouterProduit(Produit p, Long idCat) {
    Categorie c=getCategorie(idCat);
    p.setCategorie(c);em.persist(p);
    return p.getIdProduit();
}
```

BoutiqueDaoImpl

@Override

```
public List<Produit> listproduits() {  
    Query req=em.createQuery("select p from Produit p");  
    return req.getResultList();  
}
```

@Override

```
public List<Produit> produitsParMotCle(String mc) {  
    Query req=em.createQuery("select p from Produit p where p.designation like  
        :x or p.description like:x");  
    req.setParameter("x", "%" + mc + "%");  
    return req.getResultList();  
}
```

@Override

```
public List<Produit> produitsParCategorie(Long idCat) {  
    Query req=em.createQuery("select p from Produit p where  
        p.categorie.idCategorie=:x");  
    req.setParameter("x", idCat);  
    return req.getResultList();  
}
```

BoutiqueDaoImpl

@Override

```
public List<Produit> produitsSelectionnees() {  
    Query req=em.createQuery("select p from Produit p where  
        p.selectionne=true");  
    return req.getResultList();  
}
```

@Override

```
public Produit getProduit(Long idP) {  
    return em.find(Produit.class, idP);  
}
```

@Override

```
public void supprimerProduit(Long idP) {  
    Produit p=getProduit(idP);  
    em.remove(p);  
}
```


BoutiqueDaoImpl

```
@Override
public void modifierProduit(Produit p) {
    em.merge(p);
}

@Override
public void ajouterUser(User u) {
    em.persist(u);
}

@Override
public void attribuerRole(Role r, Long userID) {
    User u=em.find(User.class, userID);
    r.setUser(u);
    em.persist(r);
}
```

BoutiqueDaolmpl

@Override

```
public Commande enregistrerCommande(Panier panier, Client c) {  
    em.persist(c);  
    Commande cmd=new Commande();  
    cmd.setClient(c);  
    cmd.setLigneCommandes(panier.getItems());  
    em.persist(cmd);  
    return cmd;  
}  
}
```



COUCHE MÉTIER

Interface

InternauteBoutiqueMetier

```
package net.youssfi.eboutique.metier;
import java.util.List;
import net.youssfi.eboutique.entities.*;
public interface InternauteBoutiqueMetier {
    public List<Categorie> listCategories();
    public Categorie getCategorie(Long idCat);
    public List<Produit> listproduits();
    public List<Produit> produitsParMotCle(String mc);
    public List<Produit> produitsParCategorie(Long idCat);
    public List<Produit> produitsSelectionnees();
    public Produit getProduit(Long idP);
    public Commande enregistrerCommande(Panier p,Client c);
}
```

Interface IAdminProduitsMetier

```
package net.youssfi.eboutique.metier;  
import net.youssfi.eboutique.entities.Produit;  
public interface IAdminProduitMetier extends  
    InternauteBoutiqueMetier {  
    public Long ajouterProduit(Produit p, Long idCat);  
    public void supprimerProduit(Long idP);  
    public void modifierProduit(Produit p);  
}
```

Interface IAdminCategoriesMetier

```
package net.youssfi.eboutique.metier;
import net.youssfi.eboutique.entities.Categorie;
import net.youssfi.eboutique.entities.Role;
import net.youssfi.eboutique.entities.User;
public interface IAdminCategoriesMetier extends IAdminProduitMetier {
    public Long ajouterCategorie(Categorie c);
    public void supprimerCategorie(Long idcat);
    public void modifierCategorie(Categorie c);
    public void ajouterUser(User u);
    public void attribuerRole(Role r, Long userID);
}
```

Implémentation BoutiqueMetierImpl

```
package net.youssfi.eboutique.metier;
import java.util.List;
import org.springframework.transaction.annotation.Transactional;
import net.youssfi.eboutique.dao.*;
import net.youssfi.eboutique.entities.*;
@Transactional
public class BoutiqueMetierImpl implements IAdminCategoriesMetier{
    private IBoutiqueDao dao;
    public void setDao(IBoutiqueDao dao) {
        this.dao = dao;
    }
    @Override
    public Long ajouterProduit(Produit p, Long idCat) {
        return dao.ajouterProduit(p, idCat);
    }
}
```

Implémentation BoutiqueMetierImpl

```
@Override
public void supprimerProduit(Long idP) {
    dao.supprimerProduit(idP);
}

@Override
public void modifierProduit(Produit p) {
    dao.modifierProduit(p);
}

@Override
public List<Categorie> listCategories() {
    return dao.listCategories();
}

@Override
public Categorie getCategorie(Long idCat) {
    return dao.getCategorie(idCat);
}
```


Implémentation BoutiqueMetierImpl

```
@Override
public List<Produit> listproduits() {
    return dao.listproduits();
}

@Override
public List<Produit> produitsParMotCle(String mc) {
    return dao.produitsParMotCle(mc);
}

@Override
public List<Produit> produitsParCategorie(Long idCat) {
    return dao.produitsParCategorie(idCat);
}

@Override
public List<Produit> produitsSelectionnes() {
    return dao.produitsSelectionnes();
}
```

Implémentation

BoutiqueMetierImpl

```
@Override
public Produit getProduit(Long idP) {
    return dao.getProduit(idP);
}

@Override
public Commande enregistrerCommande(Panier p, Client c) {
    return dao.enregistrerCommande(p, c);
}

@Override
public Long ajouterCategorie(Categorie c) {
    return dao.ajouterCategorie(c);
}

@Override
public void supprimerCategorie(Long idcat) {
    dao.supprimerCategorie(idcat);
}
```

Implémentation BoutiqueMetierImpl

```
@Override
public void modifierCategorie(Categorie c) {
    dao.modifierCategorie(c);
}

@Override
public void ajouterUser(User u) {
    dao.ajouterUser(u);
}

@Override
public void attribuerRole(Role r, Long userID) {
    dao.attribuerRole(r, userID);
}
}
```



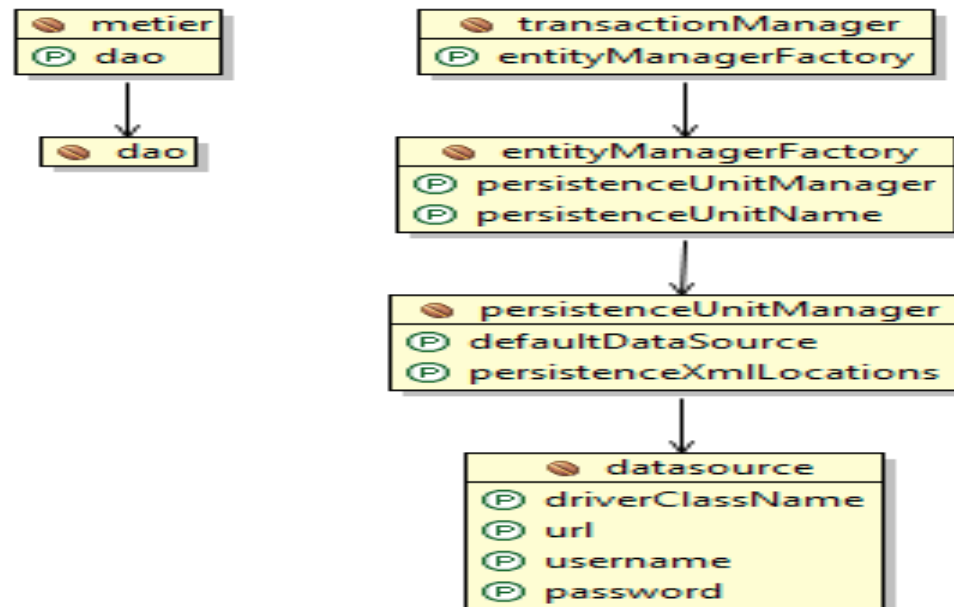
Tester les deux couches métiers et dao

Injection des dépendances :

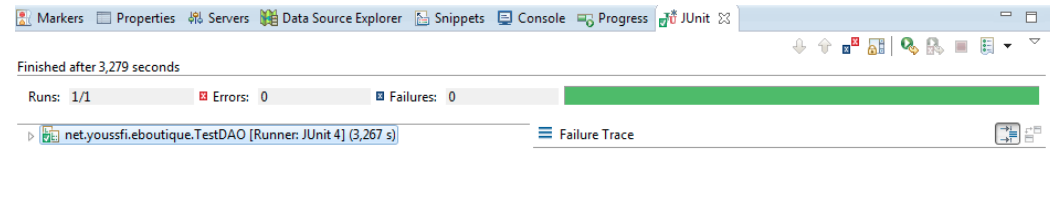
/EBoutiqueV2/src/main/resources/applicationContext.xml

```
<bean id="dao" class="net.youssfi.eboutique.dao.BoutiqueDaoImpl"></bean>
<bean id="metier" class="net.youssfi.eboutique.metier.BoutiqueMetierImpl">
  <property name="dao" ref="dao"></property>
</bean>
```

Spring Beans



JUnit Test



```
package net.youssfi.eboutique;

import static org.junit.Assert.*;import java.util.List;import net.youssfi.eboutique.entities.*;
import net.youssfi.eboutique.metier.IAdminCategoriesMetier;
import org.junit.*; import org.springframework.context.support.*;

public class TestDAO {

    @Before

    public void setUp() throws Exception { }

    @Test

    public void test() {
        try {
            ClassPathXmlApplicationContext context=
new ClassPathXmlApplicationContext(new String[]{"applicationContext.xml"});
            IAdminCategoriesMetier metier=(IAdminCategoriesMetier) context.getBean("metier");
            List<Categorie> cats1=metier.listCategories();
            metier.ajouterCategorie(new Categorie("Ordinateur", "Ordinateurs", "", null));
            metier.ajouterCategorie(new Categorie("Imprimantes", "Imprimantes", "", null));
            List<Categorie> cats2=metier.listCategories();
            assertTrue(cats2.size()==cats1.size()+2);
        } catch (Exception e) { assertTrue(e.getMessage(),false);}}
    }
}
```



COUCHE WEB

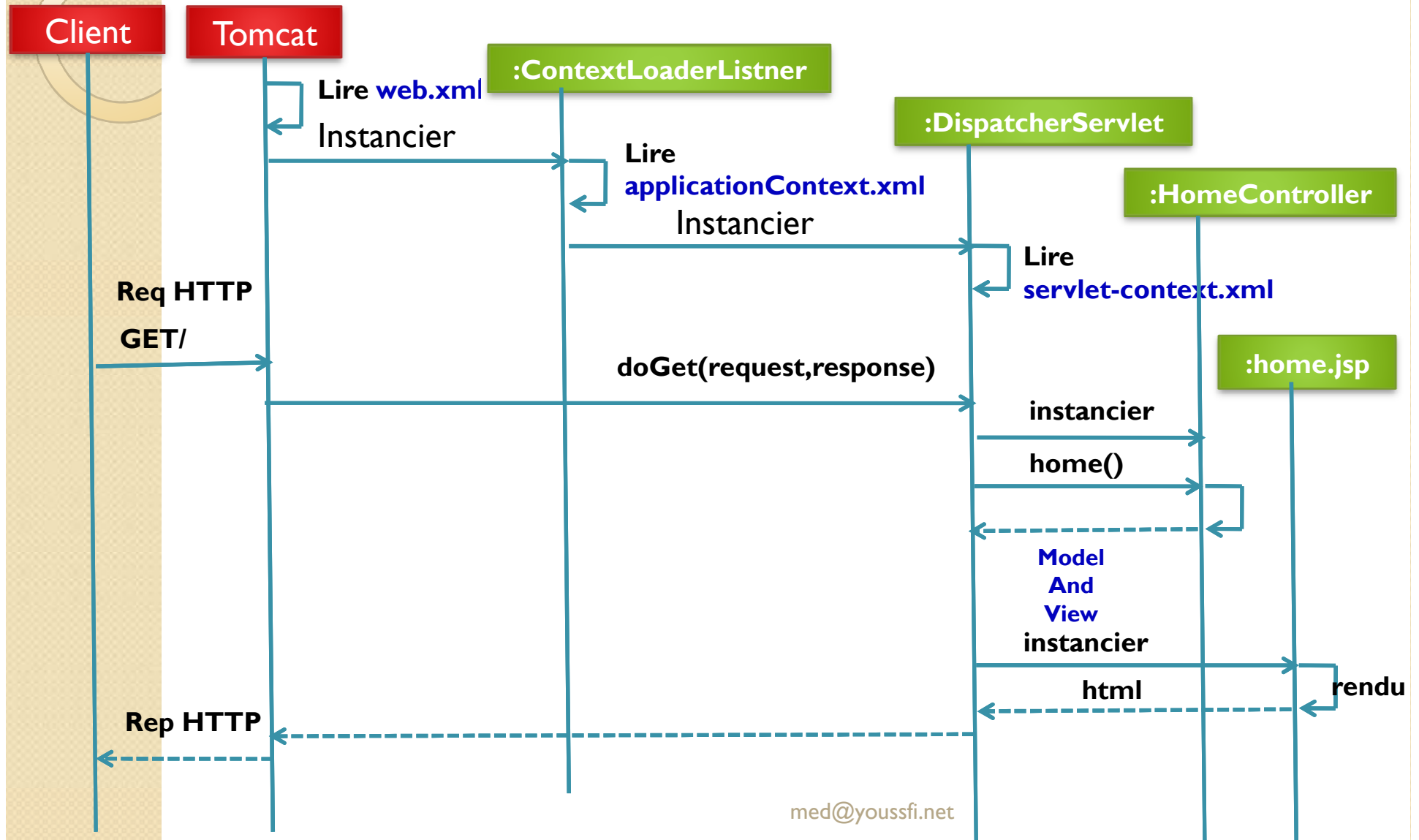
web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-
app_2_5.xsd">
<!-- The definition of the Root Spring Container shared by all Servlets and Filters -
->
<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>classpath*:applicationContext.xml</param-value>
</context-param>
<!-- Creates the Spring Container shared by all Servlets and Filters -->
<listener>
    <listener-class>
org.springframework.web.context.ContextLoaderListener
    </listener-class>
</listener>
```


web.xml

```
<!-- Processes application requests -->
<servlet>
  <servlet-name>appServlet</servlet-name>
  <servlet-
class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  <init-param>
<param-name>contextConfigLocation</param-name>
<param-value>/WEB-INF/spring/appServlet/servlet-context.xml</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>appServlet</servlet-name>
  <url-pattern>/</url-pattern>
</servlet-mapping>
</web-app>
```

Fonctionnement



Maven Dependencies

```
<!-- Apache Commons Upload -->
<dependency>
  <groupId>commons-fileupload</groupId>
  <artifactId>commons-fileupload</artifactId>
  <version>1.2.2</version>
</dependency>
<dependency>
  <groupId>org.apache.commons</groupId>
  <artifactId>commons-io</artifactId>
  <version>1.3.2</version>
</dependency>
```

/WEB-INF/spring/appServlet/servlet-context.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans:beans xmlns="http://www.springframework.org/schema/mvc"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:beans="http://www.springframework.org/schema/beans"
xmlns:context="http://www.springframework.org/schema/context"
xsi:schemaLocation="http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc.xsd
http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-
beans.xsd
http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-
context.xsd">
<!-- DispatcherServlet Context: defines this servlet's request-
processing infrastructure -->
<!-- Enables the Spring MVC @Controller programming model -->
<annotation-driven />
<!-- Handles HTTP GET requests for /resources/** by efficiently
serving up static resources in the ${webappRoot}/resources
directory -->
<resources mapping="/resources/**" location="/resources/" />
```

/WEB-INF/spring/appServlet/servlet-context.xml

```
<beans:bean
    class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <beans:property name="prefix" value="/WEB-INF/views/" />
    <beans:property name="suffix" value=".jsp" />
</beans:bean>
<context:component-scan base-package="net.youssfi.eboutique" />
</beans:beans>

<context:component-scan base-package="net.youssfi.eboutique" />

<!-- Configuration Upload-->

<beans:bean name="multipartResolver"
    class="org.springframework.web.multipart.commons.CommonsMultipartResolver">
    <beans:property name="maxUploadSize" value="100000"></beans:property>
</beans:bean>
```



Contrôleurs

AdminCategoriesController

```
package net.youssfi.eboutique.controllers; import java.io.*;
import javax.validation.Valid;
import net.youssfi.eboutique.entities.Categorie;
import net.youssfi.eboutique.metier.IAdminCategoriesMetier;
import org.apache.commons.io.IOUtils;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.MediaType;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;
import org.springframework.web.bind.annotation.SessionAttributes;
import org.springframework.web.multipart.MultipartFile;
```

AdminCategoriesController

```
@Controller
@RequestMapping("/adminCat")
@SessionAttributes("editedCat")
public class AdminCategoriesController {
    @Autowired
    private IAdminCategoriesMetier metier;

    @RequestMapping("/index")
    public String index(Model model){
        model.addAttribute("categorie", new Categorie());
        model.addAttribute("categories", metier.listCategories());
        return "categories";
    }
}
```


AdminCategoriesController

```
@RequestMapping("/saveCat")
public String saveCat(@Valid Categorие c, BindingResult bindingResult,
    Model model, MultipartFile file) throws Exception{
    if(bindingResult.hasErrors()){
        model.addAttribute("categories", metier.listCategories());
        return "categories";
    }
    if(!file.isEmpty()){c.setPhoto(file.getBytes());}
    else{
        if(c.getIdCategorie()!=null){
            Categorие cat=(Categorие) model.asMap().get("editedCat");
            c.setPhoto(cat.getPhoto());
        }
        if(c.getIdCategorie()==null) metier.ajouterCategorie(c);
        else metier.modifierCategorie(c);
        model.addAttribute("categorie", new Categorие());
        model.addAttribute("categories", metier.listCategories());
        return "categories";
    }
}
```

AdminCategoriesController

```
@RequestMapping(value="/photoCat",produces=MediaType.IMAGE_JPEG_VALUE)
```

```
@ResponseBody
```

```
public byte[] getPhoto(Long idCat) throws IOException{  
    Categorie c=metier.getCategorie(idCat);  
    if(c.getPhoto()==null) return new byte[0];  
    else return IOUtils.toByteArray(new ByteArrayInputStream(c.getPhoto()));  
}
```

```
@RequestMapping(value="/suppCat")
```

```
public String suppCat(Long idCat,Model model){  
    metier.supprimerCategorie(idCat);  
    model.addAttribute("categorie", new Categorie());  
    model.addAttribute("categories", metier.listCategories());  
    return "categories";  
}
```

```
@RequestMapping(value="/editCat")
```


```
public String editCat(Long idCat,Model model){  
    Categorie c=metier.getCategorie(idCat);  
    model.addAttribute("editedCat", c);model.addAttribute("categorie",c );  
    model.addAttribute("categories", metier.listCategories());  
    return "categories";} }
```

Vue : categories.jsp

localhost / localhost / ebc x localhost:8080/eboutique x

localhost:8080/eboutique/adminCat/saveCat

ID Catégorie:
Nom Catégorie
Description
Photo Aucun fichier choisi

ID	NOM CAT	DESCRIPTION	PHOTO		
11	Ordinateurs	mmmmmmmmmm		Supprimer	Edit
12	aaaaaaa	mmmmmm		Supprimer	Edit

med@yousii.net

Vue : categories.jsp

```
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@taglib uri="http://www.springframework.org/tags/form" prefix="f"%>
<head>
  <link rel="stylesheet" type="text/css" href="<%=request.getContextPath()
    %>/resources/css/style.css">
</head>
<div id="formCat" class="cadre">
  <f:form modelAttribute="categorie" action="saveCat" method="post"
    enctype="multipart/form-data">
<table>
  <tr>
    <td>ID Catégorie:</td>
    <td>${categorie.idCategorie}<f:input type="hidden" path="idCategorie"/></td>
    <td><f:errors path="idCategorie"></f:errors> </td>
  </tr>
  <tr>
    <td>Nom Catégorie</td><td><f:input path="nomCategorie"/></td>
    <td><f:errors path="nomCategorie"></f:errors> </td>
  </tr>
```

Vue : categories.jsp

```
<tr>
  <td>Description</td>
  <td><f:textarea path="description"/></td>
  <td><f:errors path="description"></f:errors> </td>
</tr>
<tr>
  <td>Photo</td>
  <c:if test="${categorie.idCategorie!=null}">
    <td></td>
  </c:if>
  <td>
    <input type="file" name="file"></td>
  </tr>
<tr>
  <td><input type="submit" value="Save"></td>
</tr>
</table>
</f:form>
</div>
```

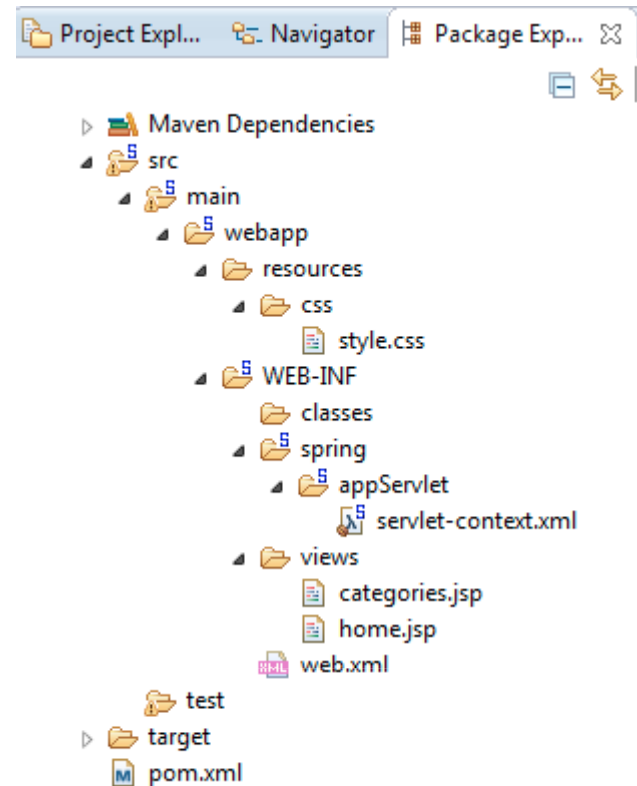
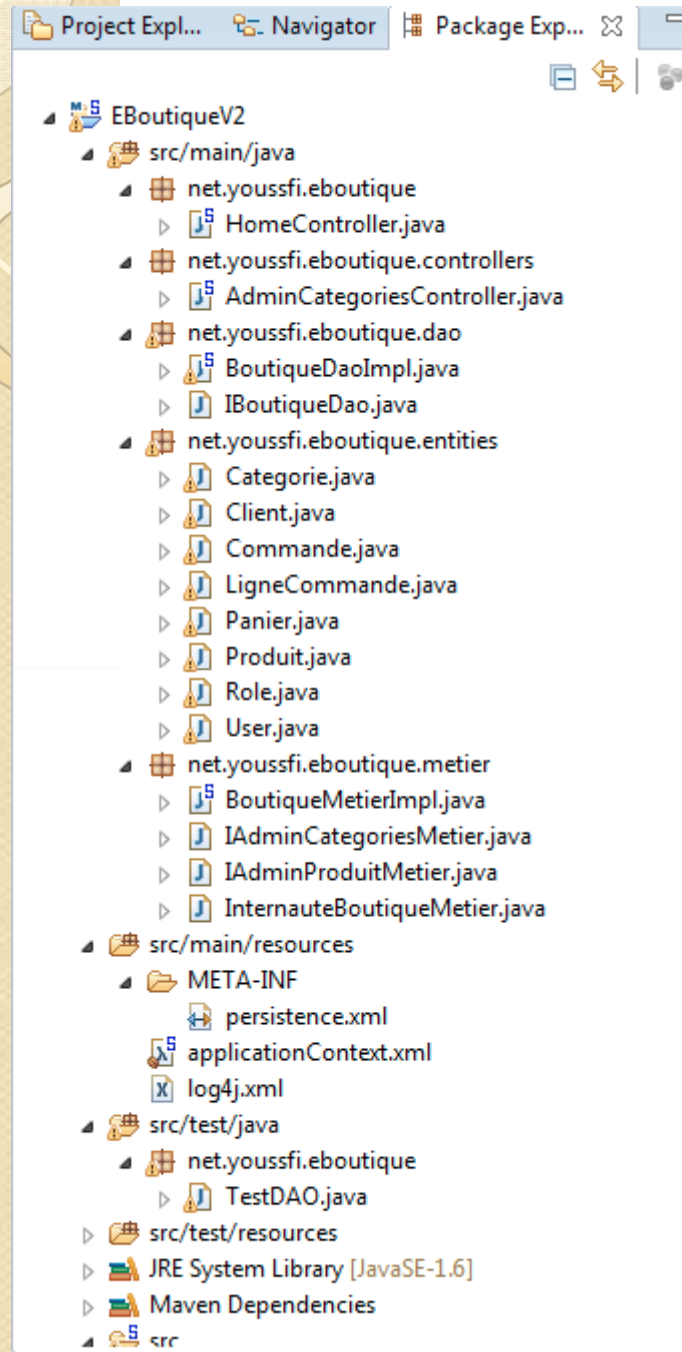
Vue : categories.jsp

```
<div id="tabCategories" class="cadre">
<table class="tabStyle1">
  <tr>
    <th>ID</th><th>NOM CAT</th><th>DESCRIPTION</th><th>PHOTO</th>
    <th></th><th></th>
  </tr>
  <c:forEach items="${categories}" var="cat">
    <tr>
      <td>${cat.idCategorie}</td>
      <td>${cat.nomCategorie}</td>
      <td>${cat.description}</td>
      <td></td>
      <td><a href="suppCat?idCat=${cat.idCategorie}">Supprimer</a></td>
      <td><a href="editCat?idCat=${cat.idCategorie}">Edit</a></td>
    </tr>
  </c:forEach>
</table>
</div>
```

Feuille de style CSS :

</EBoutiqueV2/src/main/webapp/resources/css/style.css>

```
div.cadre{  
    border: 1px dotted gray;  
    border-radius:10px;  
    margin: 10px;  
    padding: 10px;  
}  
.tabStyle1 th{  
    border: 1px dotted gray;  
    background: pink;  
    padding: 5px;  
}  
.tabStyle1 td{  
    border: 1px dotted gray;  
    background: white;  
    padding: 5px;  
}
```



Structure du projet



SPRING SECURITY

Maven Dependencies

```
<!-- Spring Security-->
```

```
<dependency>
```

```
  <groupId>org.springframework.security</groupId>
```

```
  <artifactId>spring-security-core</artifactId>
```

```
  <version>3.2.0.RELEASE</version>
```

```
</dependency>
```

```
<dependency>
```

```
  <groupId>org.springframework.security</groupId>
```

```
  <artifactId>spring-security-config</artifactId>
```

```
  <version>3.2.0.RELEASE</version>
```

```
</dependency>
```

```
<dependency>
```

```
  <groupId>org.springframework.security</groupId>
```

```
  <artifactId>spring-security-web</artifactId>
```

```
  <version>3.2.0.RELEASE</version>
```

```
</dependency>
```

web.xml

- Déclarer le filtre DelegatingFilterProxy dans le fichier web.xml
- Toutes les requêtes HTTP passent par ce filtre.
- Le nom du filtre est : **springSecurityFilterChain**
- Ce nom devrait correspondre au nom d'un bean spring qui sera déployé par ContextLoaderListener et qui contient les règles de sécurité à exécuter.

```
<!-- Spring Security -->
<filter>
    <filter-name>springSecurityFilterChain</filter-name>
    <filter-
class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
</filter>
<filter-mapping>
    <filter-name>springSecurityFilterChain</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

Base de données

- Créer deux tables :
 - Users : qui contient les utilisateurs autorisés à accéder à l'application
 - Roles : qui contient les rôles de chaque utilisateur

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra
1	<u>ID_USER</u>	int(11)			Non	Aucune	AUTO_INCREMENT
2	username	varchar(15)	latin1_swedish_ci		Non	Aucune	
3	PASSWORD	varchar(100)	latin1_swedish_ci		Non	Aucune	
4	ACTIVED	tinyint(1)			Non	Aucune	

Nom	Type	Interclassement	Attributs	Null	Défaut	Extra
<u>ID_ROLE</u>	int(11)			Non	Aucune	AUTO_INCREMENT
ID_USER	int(11)			Non	Aucune	
ROLE_NAME	varchar(20)	latin1_swedish_ci		Non	Aucune	

ID_USER	username	PASSWORD	ACTIVED
1	admin1	e00cf25ad42683b3df678c61f42c6bda	1
2	admin2	c84258e9c39059a89ab77d846ddab909	1
3	user	ee11cbb19052e40b07aac0ca060c23ee	1

ID_ROLE	ID_USER	ROLE_NAME
1	1	ROLE_ADMIN_CAT
2	1	ROLE_ADMIN_PROD
3	2	ROLE_ADMIN_PROD
4	3	ROLE_USER

Base de données : Table Users

```
--  
-- Structure de la table `users`  
--  
CREATE TABLE IF NOT EXISTS `users` (  
  `ID_USER` int(11) NOT NULL AUTO_INCREMENT,  
  `username` varchar(15) NOT NULL,  
  `PASSWORD` varchar(100) NOT NULL,  
  `ACTIVED` tinyint(1) NOT NULL,  
  PRIMARY KEY (`ID_USER`),  
  UNIQUE KEY `LOGIN` (`username`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;  
--  
-- Contenu de la table `users`  
--  
INSERT INTO `users` (`ID_USER`, `username`, `PASSWORD`, `ACTIVED`) VALUES  
(1, 'admin1', 'e00cf25ad42683b3df678c61f42c6bda', 1),  
(2, 'admin2', 'c84258e9c39059a89ab77d846ddab909', 1),  
(3, 'user', 'ee11cbb19052e40b07aac0ca060c23ee', 1);
```

Base de données : Table roles

```
--  
-- Structure de la table `roles`  
--  
CREATE TABLE IF NOT EXISTS `roles` (  
  `ID_ROLE` int(11) NOT NULL AUTO_INCREMENT,  
  `ID_USER` int(11) NOT NULL,  
  `ROLE_NAME` varchar(20) NOT NULL,  
  PRIMARY KEY (`ID_ROLE`),  
  KEY `ID_USER` (`ID_USER`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;  
--  
-- Contenu de la table `roles`  
--  
INSERT INTO `roles` (`ID_ROLE`, `ID_USER`, `ROLE_NAME`) VALUES  
(1, 1, 'ROLE_ADMIN_CAT'),  
(2, 1, 'ROLE_ADMIN_PROD'),  
(3, 2, 'ROLE_ADMIN_PROD'),  
(4, 3, 'ROLE_USER');  
--  
-- Contraintes pour la table `roles`  
--  
ALTER TABLE `roles`  
  ADD CONSTRAINT `roles_ibfk_1` FOREIGN KEY (`ID_USER`) REFERENCES `users` (`ID_USER`);
```

applicationContext.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:tx="http://www.springframework.org/schema/tx"
xmlns:context="http://www.springframework.org/schema/context"
xmlns:s="http://www.springframework.org/schema/security"
xsi:schemaLocation="http://www.springframework.org/schema/security
    http://www.springframework.org/schema/security/spring-security-3.2.xsd
http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-3.2.xsd
http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.2.xsd">

    <bean id="datasource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
        <property name="driverClassName" value="com.mysql.jdbc.Driver"></property>
        <property name="url" value="jdbc:mysql://localhost:3306/DB_SID_CAT"></property>
        <property name="username" value="root"></property>
        <property name="password" value=""></property>
    </bean>
```

applicationContext.xml

```
<s:http>
  <s:intercept-url pattern="/produits/**" access="ROLE_ADMIN_PROD"/>
  <s:intercept-url pattern="/categories/**" access="ROLE_ADMIN_CAT"/>
  <s:form-login login-page="/login" default-target-url="/produits/index"
    authentication-failure-url="/login" />
  <s:logout logout-success-url="/logout" />
</s:http>
<s:authentication-manager>
  <s:authentication-provider>
    <s:password-encoder hash="md5"></s:password-encoder>
    <s:jdbc-user-service data-source-ref="datasource"
      users-by-username-query="select username,password, actived
        from users where username=?"
      authorities-by-username-query="select u.username, r.role_name from users u, roles r
        where u.id_user = r.id_user and u.username =? " />
    <!--
      <s:user-service>
        <s:user name="admin1" password="admin1" authorities="ROLE_ADMIN_PROD"/>
        <s:user name="admin2" authorities="ROLE_ADMIN_CAT,ROLE_ADMIN_PROD" password="admin2" />
      </s:user-service>
    -->
  </s:authentication-provider>
</s:authentication-manager>
</beans>
```


Contrôleur

```
package org.enset.sid.controllers;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
public class LoginController {
    @RequestMapping("/login")
    public String login(){
        return "login";
    }
}
```

login.jsp

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
  <form action="j_spring_security_check" method="post">
    <table>
      <tr>
        <td>Login</td>
        <td><input type="text" name="j_username"></td>
      </tr>
      <tr>
        <td>Pass word</td>
        <td><input type="password" name="j_password"></td>
      </tr>
      <tr>
        <td><input type="submit" value="Login"></td>
      </tr>
    </table>
  </form>
</body>
</html>
```

Maven Dependencies

```
<!-- Logging -->
<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>slf4j-api</artifactId>
  <version>${org.slf4j-version}</version>
</dependency>
<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>jcl-over-slf4j</artifactId>
  <version>${org.slf4j-version}</version>
  <scope>runtime</scope>
</dependency>
<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>slf4j-log4j12</artifactId>
  <version>${org.slf4j-version}</version>
  <scope>runtime</scope>
</dependency>
```

Maven Dependencies

```
<!-- Log4j -->
<dependency>
  <groupId>log4j</groupId>
  <artifactId>log4j</artifactId>
  <version>1.2.15</version>
  <exclusions>
    <exclusion>
      <groupId>javax.mail</groupId>
      <artifactId>mail</artifactId>
    </exclusion>
    <exclusion>
      <groupId>javax.jms</groupId>
      <artifactId>jms</artifactId>
    </exclusion>
    <exclusion>
      <groupId>com.sun.jdmk</groupId>
      <artifactId>jmxtools</artifactId>
    </exclusion>
    <exclusion>
      <groupId>com.sun.jmx</groupId>
      <artifactId>jmxri</artifactId>
    </exclusion>
  </exclusions>
  <scope>runtime</scope>
</dependency>
```

Maven Dependencies

```
<!-- Servlet -->
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>servlet-api</artifactId>
  <version>2.5</version>
  <scope>provided</scope>
</dependency>
<dependency>
  <groupId>javax.servlet.jsp</groupId>
  <artifactId>jsp-api</artifactId>
  <version>2.1</version>
  <scope>provided</scope>
</dependency>
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>jstl</artifactId>
  <version>1.2</version>
</dependency>
```

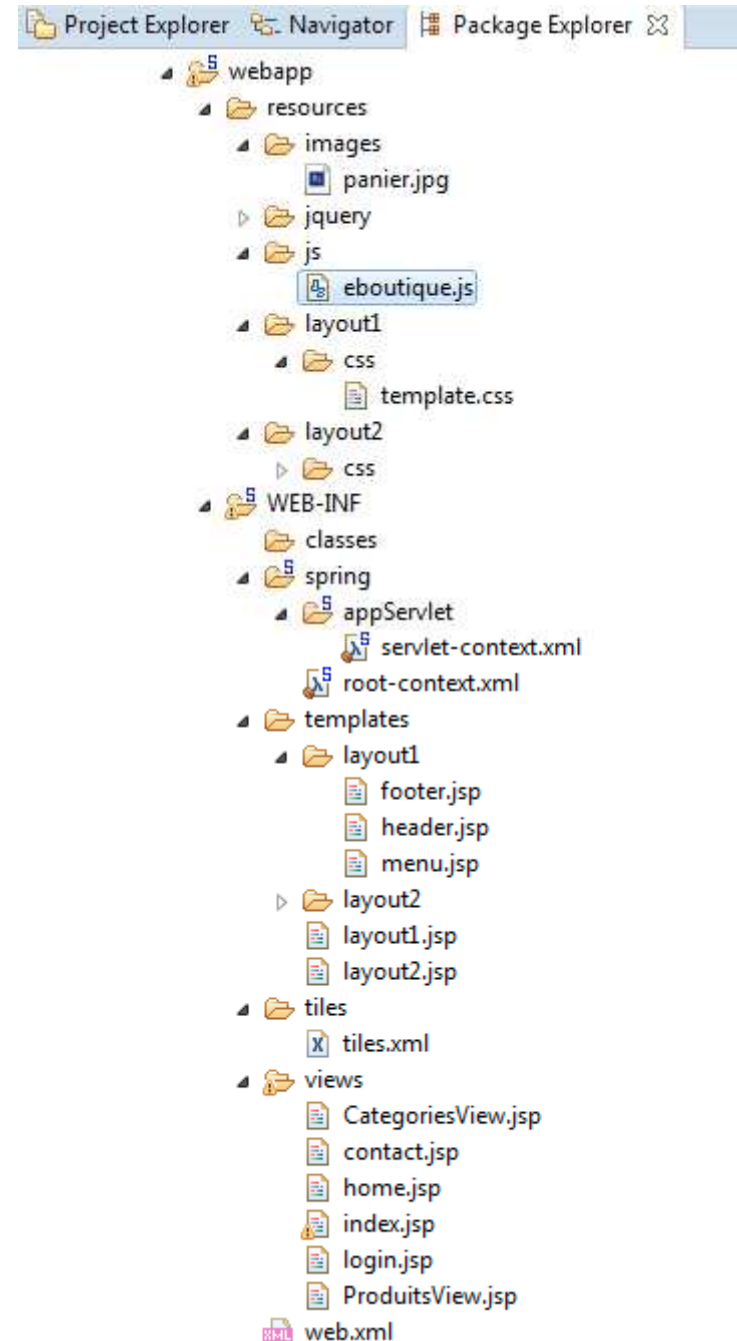
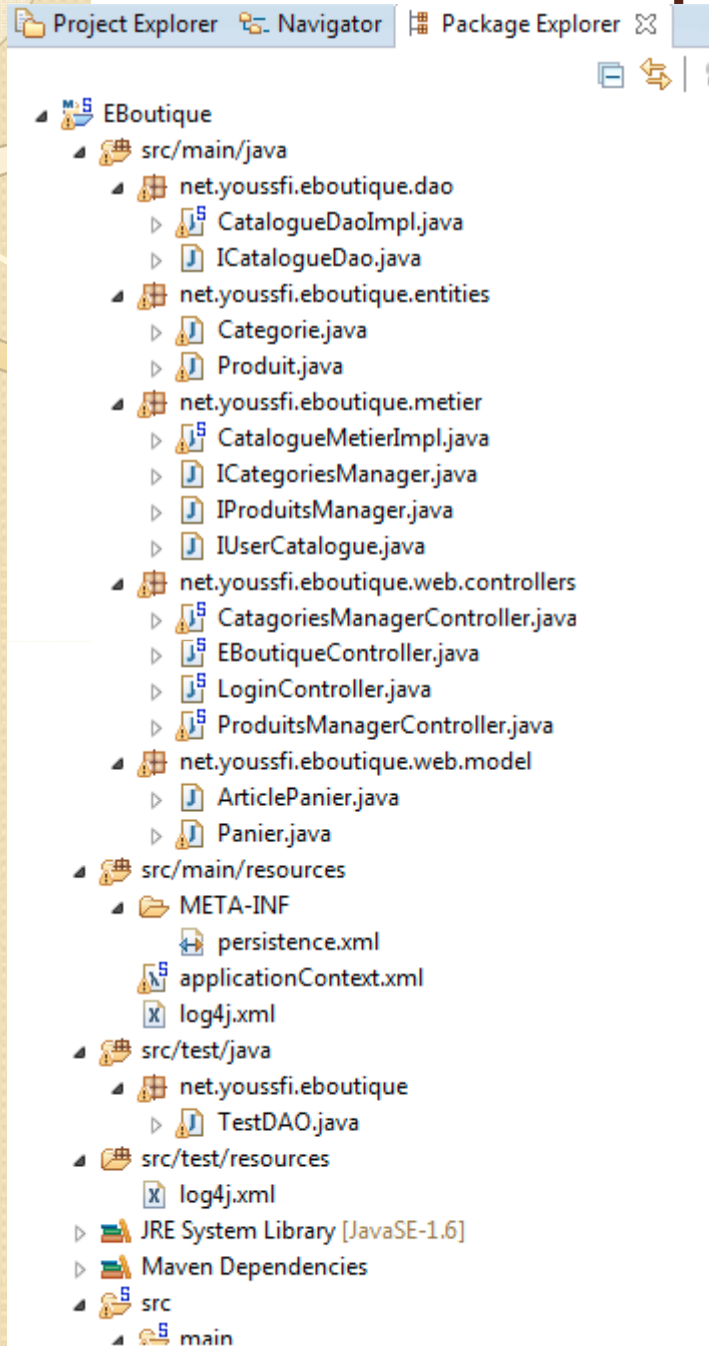
Maven Dependencies

```
<!-- AspectJ -->
<dependency>
  <groupId>org.aspectj</groupId>
  <artifactId>aspectjrt</artifactId>
  <version>${org.aspectj-version}</version>
</dependency>
<!-- @Inject -->
<dependency>
  <groupId>javax.inject</groupId>
  <artifactId>javax.inject</artifactId>
  <version>1</version>
</dependency>
<!-- Test -->
<dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <version>4.7</version>
  <scope>test</scope>
</dependency>
```

Maven Dependencies

```
<!-- Tiles 2 Dependencies -->
<dependency>
  <groupId>org.apache.tiles</groupId>
  <artifactId>tiles-api</artifactId>
  <version>2.1.4</version>
</dependency>
<dependency>
  <groupId>org.apache.tiles</groupId>
  <artifactId>tiles-core</artifactId>
  <version>2.1.4</version>
</dependency>
<dependency>
  <groupId>org.apache.tiles</groupId>
  <artifactId>tiles-jsp</artifactId>
  <version>2.1.4</version>
</dependency>
<dependency>
  <groupId>org.apache.tiles</groupId>
  <artifactId>tiles-servlet</artifactId>
  <version>2.1.4</version>
</dependency>
```

Structure du projet



web.xml

```
<!-- Spring Security -->
<filter>
    <filter-name>springSecurityFilterChain</filter-name>
    <filter-
class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
</filter>
<filter-mapping>
    <filter-name>springSecurityFilterChain</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
<!-- The definition of the Root Spring Container shared by all Servlets and
Filters -->
<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>classpath*:applicationContext.xml</param-value>
</context-param>
```

web.xml

```
<!-- Creates the Spring Container shared by all Servlets and Filters -->
<listener>
  <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>
<!-- Processes application requests -->
<servlet>
  <servlet-name>appServlet</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  <init-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/spring/appServlet/servlet-context.xml</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>appServlet</servlet-name>
  <url-pattern>/</url-pattern>
</servlet-mapping>
<error-page>
  <error-code>403</error-code>
  <location>/login</location>
</error-page>
```

servlet-context.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns:mvc="http://www.springframework.org/schema/mvc"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.springframework.org/schema/beans"
xmlns:context="http://www.springframework.org/schema/context"
xsi:schemaLocation="http://www.springframework.org/schema/mvc
    http://www.springframework.org/schema/mvc/spring-mvc.xsd
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context.xsd">
<!-- DispatcherServlet Context: defines this servlet's request-processing
    infrastructure -->
<!-- Enables the Spring MVC @Controller programming model -->
<mvc:annotation-driven />
<!-- Handles HTTP GET requests for /resources/** by efficiently serving up
    static resources in the ${webappRoot}/resources directory -->
<mvc:resources mapping="/resources/**" location="/resources/" />
<context:component-scan base-package="net.youssfi.eboutique" />
```

servlet-context.xml

```
<!-- Tiles View Resolver Configuration -->
<bean id="tilesViewresolver"
    class="org.springframework.web.servlet.view.UrlBasedViewResolver">
    <property
        name="viewClass"
        value="org.springframework.web.servlet.view.tiles2.TilesView">
    </property>
</bean>
<bean id="tilesConfigurer"
    class="org.springframework.web.servlet.view.tiles2.TilesConfigurer">
    <property name="definitions">
        <value>/WEB-INF/tiles/tiles.xml</value>
    </property>
</bean>
<bean id="multipartResolver"
    class="org.springframework.web.multipart.commons.CommonsMultipartResolver">
    <property name="maxUploadSize" value="10000000"></property>
</bean>

</beans>
```

/WEB-INF/tiles/tiles.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE tiles-definitions PUBLIC
    "-//Apache Software Foundation//DTD Tiles Configuration 2.0//EN"
    "http://tiles.apache.org/dtds/tiles-config_2_0.dtd">
<tiles-definitions>
    <definition name="layout1" template="/WEB-INF/templates/layout1.jsp">
        <put-attribute name="title" value=""></put-attribute>
        <put-attribute name="header" value="/WEB-INF/templates/layout1/header.jsp"></put-attribute>
        <put-attribute name="menu" value="/WEB-INF/templates/layout1/menu.jsp"></put-attribute>
        <put-attribute name="body" value=""></put-attribute>
        <put-attribute name="footer" value="/WEB-INF/templates/layout1/footer.jsp"></put-attribute>
    </definition>
    <definition name="layout2" template="/WEB-INF/templates/layout2.jsp">
        <put-attribute name="title" value=""></put-attribute>
        <put-attribute name="header" value="/WEB-INF/templates/layout2/header.jsp"></put-attribute>
        <put-attribute name="menu" value="/WEB-INF/templates/layout2/menu.jsp"></put-attribute>
        <put-attribute name="body" value=""></put-attribute>
        <put-attribute name="footer" value="/WEB-INF/templates/layout2/footer.jsp"></put-attribute>
    </definition>
</tiles-definitions>
```

/WEB-INF/tiles/tiles.xml

```
<definition name="ProduitsView" extends="layout1">
    <put-attribute name="title" value="Produits" />
    <put-attribute name="body" value="/WEB-INF/views/ProduitsView.jsp" />
</definition>
<definition name="CategoriesView" extends="layout1">
    <put-attribute name="title" value="Catégories" />
    <put-attribute name="body" value="/WEB-INF/views/CategoriesView.jsp" />
</definition>
<definition name="Login" extends="layout1">
    <put-attribute name="title" value="Login" />
    <put-attribute name="body" value="/WEB-INF/views/login.jsp" />
</definition>
<definition name="index" extends="layout2">
    <put-attribute name="title" value="E Boutique" />
    <put-attribute name="body" value="/WEB-INF/views/index.jsp" />
</definition>
</tiles-definitions>
```

/WEB-INF/templates/Layout1.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@taglib uri="http://tiles.apache.org/tags-tiles" prefix="tiles" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
<link rel="stylesheet" type="text/css"
    href="<%=request.getContextPath()%>/resources/layout1/css/template.css"/>
</head>
<body>
    <div id="header" class="cadre">
        <tiles:insertAttribute name="header"></tiles:insertAttribute>
    </div>
    <div id="content" class="cadre">
        <div id="menu" class="cadre"><tiles:insertAttribute name="menu"></tiles:insertAttribute></div>
        <div id="body" class="cadre"><tiles:insertAttribute name="body"></tiles:insertAttribute></div>
    </div>
    <div id="footer" class="cadre"><tiles:insertAttribute name="footer"></tiles:insertAttribute></div>
</body>
</html>
```

/WEB-INF/templates/Layout1/menu.jsp

```
<table>
  <tr>
    <td>
      <a href="<%=request.getContextPath()%>/categories/index">
        Categories</a></td>
    </tr>
    <tr>
      <td>
        <a href="<%=request.getContextPath()%>/produits/index">
          Produits</a></td>
      </tr>
    </table>
```


/WEB-INF/templates/Layout1/header.jsp

HEADER

/WEB-INF/templates/Layout1/footer.jsp

FOOTER

/resources/Layout1/css/template.css

```
body{
  font-family: sans-serif;
  font-size: 12px;
}
.cadre{
  margin: 0;
  border: 1px dotted gray;
  margin-top: 10px;
  padding: 10px;
}
#menu{
  width: 200px;
  min-height: 400px;
  background: #FFF;
  float: left;
  margin-left: 0;
  margin-top: 10px;
  overflow: auto;
}
```

```
#content{
  margin: 0px;
}
#body{
  min-height: 400px;
  background: #FFF;
  overflow: auto;
}
#footer{
  background: #FFF;
}
.table1{
}
.table1 th{
  padding: 5px;
  background: pink;
  border: 1px dotted gray;
}
.table1 td{
  border: 1px dotted gray;
  padding: 5px;
  background: white;
}
.errors{
  color: red;
}
```

/WEB-INF/templates/Layout2.jsp

```
<%@taglib uri="http://tiles.apache.org/tags-tiles" prefix="tiles" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
<link rel="stylesheet" type="text/css"
      href="<%=request.getContextPath()%>/resources/Layout2/css/template.css"/>
<script type="text/javascript" src="<%=request.getContextPath()%>/resources/jquery/jquery-
1.8.2.js"></script>
<script type="text/javascript" src="<%=request.getContextPath()%>/resources/js/eboutique.js"></script>
</head>
<body>
  <div id="header" class="cadre">
    <tiles:insertAttribute name="header"></tiles:insertAttribute>
  </div>
  <div id="content" class="cadre">
    <div id="menu" class="cadre"><tiles:insertAttribute name="menu"></tiles:insertAttribute></div>
    <div id="body" class="cadre"><tiles:insertAttribute name="body"></tiles:insertAttribute></div>
  </div>
  <div id="footer" class="cadre"><tiles:insertAttribute name="footer"></tiles:insertAttribute></div>
</body>
</html>
```

/WEB-INF/templates/Layout2/menu.jsp

```
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<table>
  <c:forEach items="${categories}" var="cat">
    <tr>
      <td>
        <a href="produitsParCat?idCat=${cat.idCategorie}">
          ${cat.nomCategorie}
        </a>
      </td>
    </tr>
  </c:forEach>
</table>
```

/WEB-INF/templates/Layout2/header.jsp

```
<div>
  <table>
    <tr>
      <td></td>
      <td></td> <td class="pan"> <div>
        <table>
          <tr>
            <td colspan="2"></td>
          </tr>
          <tr>
            <td>Nombre de produits :</td> <td>${panier.size}</td>
          </tr>
          <tr>
            <td>Total :</td> <td>${panier.total}</td>
          </tr>
        </table>
      </div>
    </td>
  </tr>
</table>
</div>
<div>
```

/WEB-INF/templates/Layout2/header.jsp

```
<table>
  <tr>
    <td>
      <div id="chercher">
        <form action="chercherProduits">
          <input type="text" name="mc" value="${mc}">
          <input type="submit" value="Chercher">
        </form>
      </div>
    </td>
    <td>
      <a href="index">Intex</a>
    </td>
  </tr>
</table>
</div>
```

/WEB-INF/templates/Layout2/footer.jsp

FOOTER

/resources/layout2/css/template.css

```
body{
font-family: sans-serif;
font-size: 12px;
}
.cadre{
margin: 0;
border: 1px dotted gray;
margin-top: 10px;
padding: 10px;
}
#menu{
width: 200px;
min-height: 400px;
background: #FFF;
float: left;
margin-left:0;
margin-top:10px;
overflow: auto;
}
#content{
margin: 0px;
}
```

```
#body{
min-height: 400px;
background: #FFF;
overflow: auto;
}
#footer{
background: #FFF;
}
.table1{
}
.table1 th{
padding: 5px;
background: pink;
border: 1px dotted gray;
}
.table1 td{
border:1px dotted gray;
padding: 5px;
background: white;
}
.errors{
color: red;
}
.ficheProduit{
border:1px dotted gray;
float:left;
height: 300px;
padding: 10px;
margin: 10px;
}
```

```
#panier{
border:1px dotted gray;
padding: 10px;
margin: 10px;
overflow: auto;
}
#panier table th{
border:1px dotted gray;
background: pink;
padding: 5px;
}
#panier table td{
border:1px dotted gray;
background: white;
padding: 5px;
}
#header table{
width: 100%;
}
#header td.pan{
text-align: right;
width: 200px;
}
#header td.pan div{
border: 1px dotted gray;
border-radius:5px;
}
#imgPanier{
cursor: pointer;
}
```




/resources/js/eboutique.js

```
function affichePanier(){  
    $("#panier").toggle();  
}
```



Couche DAO

Entity : Categorie

```
package net.youssfi.eboutique.entities;
import java.io.Serializable;import java.util.*;
import javax.persistence.*;import javax.validation.constraints.Size;
import org.hibernate.validator.constraints.NotEmpty;
@Entity
public class Categorie implements Serializable {
@Id
@GeneratedValue
    private Long idCategorie;
@NotEmpty
@Size(min=4,max=20)
    private String nomCategorie;
    private String description;
    private String nomPhoto;
    @Lob
    private byte[] photo;
    @OneToMany(mappedBy="categorie")
    private Collection<Produit> produits=new ArrayList<Produit>();
    // Constructeurs
    // Getters et Setters
}
```

Entity : Produit

```
package net.youssfi.eboutique.entities;
import java.io.Serializable;import javax.persistence.*;
import org.hibernate.validator.constraints.NotEmpty;
@Entity
public class Produit implements Serializable {
    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private Long idProduit;
    @NotEmpty
    private String designation;
    private String description;
    private double prix;
    private String photo;
    private int quantite;
    private boolean selectionne;
    @ManyToOne
    @JoinColumn(name="ID_CAT")
    private Categorie categorie;
    // Constructeurs
    // Getters et Setters
}
```

Interface DAO

```
package net.youssfi.eboutique.dao;
import java.util.List;
import net.youssfi.eboutique.entities.Categorie;
import net.youssfi.eboutique.entities.Produit;
public interface ICatalogueDao {
    public Long ajouterCategorie(Categorie c);
    public List<Categorie> listCategories();
    public Categorie getCategorie(Long idCat);
    public void supprimerCategorie(Long idcat);
    public void modifierCategorie(Categorie c);
    public Long ajouterProduit(Produit p, Long idCat);
    public List<Produit> listproduits();
    public List<Produit> produitsParMotCle(String mc);
    public List<Produit> produitsParCategorie(Long idCat);
    public List<Produit> produitsSelectionnees();
    public Produit getProduit(Long idP);
    public void supprimerProduit(Long idP);
    public void modifierProduit(Produit p);
}
```

Implémentation JPA

```
package net.youssfi.eboutique.dao;
import java.util.List;import javax.persistence.*;
import net.youssfi.eboutique.entities.Categorie;
import net.youssfi.eboutique.entities.Produit;
public class CatalogueDaoImpl implements ICatalogueDao {
    @PersistenceContext
    private EntityManager em;
    @Override
    public Long ajouterCategorie(Categorie c) {
        em.persist(c);return c.getIdCategorie();
    }
    @Override
    public List<Categorie> listCategories() {
        Query req=em.createQuery("select c from Categorie c");
        return req.getResultList();
    }
}
```

Implémentation JPA

@Override

```
public Long ajouterProduit(Produit p, Long idCat) {  
    Categorie c =getCategorie(idCat);  
    em.persist(p);  
    return p.getIdProduit();  
}
```

@Override

```
public List<Produit> listproduits() {  
    Query req=em.createQuery("select p from Produit p");  
    return req.getResultList();  
}
```

@Override

```
public List<Produit> produitsParMotCle(String mc) {  
    Query req=em.createQuery("select p from Produit p where p.designation like  
        :x or p.description like :x");  
    req.setParameter("x", "%" +mc+"%");  
    return req.getResultList();  
}
```

Implémentation JPA

@Override

```
public List<Produit> produitsParCategorie(Long idCat) {  
    Query req=em.createQuery("select p from Produit p where  
        p.categorie.idCategorie=:x");  
    req.setParameter("x", idCat);  
    return req.getResultList();  
}
```

@Override

```
public List<Produit> produitsSelectionnees() {  
    Query req=em.createQuery("select p from Produit p where  
        p.selectionne=true");  
    return req.getResultList();  
}
```

@Override

```
public void supprimerProduit(Long idP) {  
    Produit p= getProduit(idP);  
    em.remove(p);  
}
```


Implémentation JPA

```
@Override
public void modifierProduit(Produit p) {
    em.merge(p);
}

@Override
public Categorie getCategorie(Long idCat) {
    return em.find(Categorie.class, idCat);
}

@Override
public void supprimerCategorie(Long idcat) {
    Categorie c=getCategorie(idcat);
    em.remove(c);
}

@Override
public void modifierCategorie(Categorie c) {
    em.merge(c);
}

@Override
public Produit getProduit(Long idP) {
    return em.find(Produit.class,idP);
}}
```



Couche Métier

Interface IUserCatalogue

```
package net.youssfi.eboutique.metier;
import java.util.List;
import net.youssfi.eboutique.entities.Categorie;
import net.youssfi.eboutique.entities.Produit;
public interface IUserCatalogue {
    public List<Categorie> listCategories();
    public Categorie getCategorie(Long idCat);
    public List<Produit> listproduits();
    public List<Produit> produitsParMotCle(String mc);
    public List<Produit> produitsSelectionnes();
    public Produit getProduit(Long idP);
    public List<Produit> produitsParCategorie(Long idCat);
}
```

Interface IProduitManager

```
package net.youssfi.eboutique.metier;  
import net.youssfi.eboutique.entities.Produit;  
public interface IProduitsManager extends IUserCatalogue {  
    public Long ajouterProduit(Produit p, Long idCat);  
    public void supprimerProduit(Long idP);  
    public void modifierProduit(Produit p);  
}
```

Interface ICategoryManager

```
package net.youssfi.eboutique.metier;

import net.youssfi.eboutique.entities.Categorie;

public interface ICategoriesManager extends IUserCatalogue {
    public Long ajouterCategorie(Categorie c);
    public void supprimerCategorie(Long idcat);
    public void modifierCategorie(Categorie c);
}
```

Implémentation métier

```
package net.youssfi.eboutique.metier;

import java.util.List; import net.youssfi.eboutique.dao.*;
import net.youssfi.eboutique.entities.*;
import org.springframework.transaction.annotation.Transactional;

@Transactional
public class CatalogueMetierImpl implements
    IProduitsManager, ICategoriesManager {
    private ICatalogueDao dao;

    public void setDao(ICatalogueDao dao) {
        this.dao = dao;
    }

    @Override
    public List<Categorie> listCategories() {
        return dao.listCategories();
    }
}
```

Implémentation métier

```
@Override
public Categorie getCategorie(Long idCat) {
    return dao.getCategorie(idCat);
}

@Override
public List<Produit> listproduits() {
    return dao.listproduits();
}

@Override
public List<Produit> produitsParMotCle(String mc) {
    return dao.produitsParMotCle(mc);
}

@Override
public List<Produit> produitsSelectionnes() {
    return dao.produitsSelectionnes();
}
```

Implémentation métier

```
@Override
public Long ajouterCategorie(Categorie c) {
    return dao.ajouterCategorie(c);
}

@Override
public void supprimerCategorie(Long idcat) {
    // TODO Auto-generated method stub
    dao.supprimerCategorie(idcat);
}

@Override
public void modifierCategorie(Categorie c) {
    // TODO Auto-generated method stub
    dao.modifierCategorie(c);
}
```


Implémentation métier

```
@Override
public Long ajouterProduit(Produit p, Long idCat) {
return dao.ajouterProduit(p, idCat);
}
```

```
@Override
public Produit getProduit(Long idP) {
// TODO Auto-generated method stub
return dao.getProduit(idP);
}
```

```
@Override
public void supprimerProduit(Long idP) {
// TODO Auto-generated method stub
dao.supprimerProduit(idP);
}
```

Implémentation métier

```
@Override
public void modifierProduit(Produit p) {
    // TODO Auto-generated method stub
    dao.modifierProduit(p);
}

@Override
public List<Produit> produitsParCategorie(Long idCat) {
    // TODO Auto-generated method stub
    return dao.produitsParCategorie(idCat);
}
}
```

persistence.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.0" xmlns="http://java.sun.com/xml/ns/persistence"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
    http://xmlns.jcp.org/xml/ns/persistence/persistence_2_0.xsd ">
  <persistence-unit name="MY_PU" transaction-type="RESOURCE_LOCAL">
    <provider>org.hibernate.ejb.HibernatePersistence</provider>
    <properties>
      <!--
        <property name="hibernate.show_sql" value="true"/>
      -->
      <property name="hibernate.hbm2ddl.auto" value="update"/>
      <property name="hibernate.dialect"
        value="org.hibernate.dialect.MySQLDialect"/>
    </properties>
  </persistence-unit>
</persistence>
```



Injection des dépendances

/EBoutique/src/main/resources/applicationContext.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:tx="http://www.springframework.org/schema/tx"
xmlns:context="http://www.springframework.org/schema/context"
xmlns:p="http://www.springframework.org/schema/p"
xmlns:s="http://www.springframework.org/schema/security"
xsi:schemaLocation="http://www.springframework.org/schema/security
    http://www.springframework.org/schema/security/spring-security-3.2.xsd
http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-
tx-3.2.xsd
http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context-3.2.xsd">
    <bean id="dao" class="net.youssfi.eboutique.dao.CatalogueDaoImpl"> </bean>
    <bean id="metier" class="net.youssfi.eboutique.metier.CatalogueMetierImpl" >
        <property name="dao" ref="dao"></property>
    </bean>
```

/EBoutique/src/main/resources/applicationContext.xml

```
<bean id="datasource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
    <property name="driverClassName" value="com.mysql.jdbc.Driver"></property>
    <property name="url" value="jdbc:mysql://localhost:3306/DB_SID_CAT"></property>
    <property name="username" value="root"></property>
    <property name="password" value=""></property>
</bean>
<bean id="persistenceUnitManager"
class="org.springframework.orm.jpa.persistenceunit.DefaultPersistenceUnitManager">
    <property name="persistenceXmlLocations">
        <list>
            <value>classpath*:META-INF/persistence.xml</value>
        </list>
    </property>
    <property name="defaultDataSource" ref="datasource"></property>
</bean>
<bean id="entityManagerFactory"
class="org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean">
    <property name="persistenceUnitManager" ref="persistenceUnitManager"></property>
    <property name="persistenceUnitName" value="MY_PU"></property>
</bean>
```

/EBoutique/src/main/resources/applicationContext.xml

```
<bean id="transactionManager" class="org.springframework.orm.jpa.JpaTransactionManager">
  <property name="entityManagerFactory" ref="entityManagerFactory"></property>
</bean>
<tx:annotation-driven transaction-manager="transactionManager"/>
<context:annotation-config></context:annotation-config>
```

/EBoutique/src/main/resources/applicationContext.xml

```
<s:http>
  <s:intercept-url pattern="/produits/**" access="ROLE_ADMIN_PROD"/>
  <s:intercept-url pattern="/categories/**" access="ROLE_ADMIN_CAT"/>
  <s:form-login login-page="/login" default-target-url="/produits/index"
    authentication-failure-url="/login" />
  <s:logout logout-success-url="/login" />
</s:http>
<s:authentication-manager>
  <s:authentication-provider>
    <s:password-encoder hash="md5"></s:password-encoder>
    <s:jdbc-user-service data-source-ref="datasource"
      users-by-username-query="select username,password, actived
        from users where username=?"
      authorities-by-username-query="select u.username, r.role_name from users u, roles r
        where u.id_user = r.id_user and u.username =? " />
    <!-- <s:user-service>
      <s:user name="admin1" password="admin1" authorities="ROLE_ADMIN_PROD"/>
      <s:user name="admin2" authorities="ROLE_ADMIN_CAT,ROLE_ADMIN_PROD"
        password="admin2" /> </s:user-service> -->
    </s:authentication-provider>
  </s:authentication-manager>
</beans>
```


Junit Test DAO

```
package net.youssfi.eboutique;
import static org.junit.Assert.*;
import org.junit.Before;
import org.junit.Test;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class TestDAO {
    @Before
    public void setUp() throws Exception {

    }
    @Test
    public void test() {
        try {
            ClassPathXmlApplicationContext app=
            new ClassPathXmlApplicationContext(new String[]{"applicationContext.xml"});
            assertTrue(true);
        } catch (Exception e) {
            assertTrue(e.getMessage(),false);
        }

    }

}
```

Gestion du panier



ArticlePanier

```
package net.youssfi.eboutique.web.model;  
import net.youssfi.eboutique.entities.Produit;  
public class ArticlePanier {  
    private Produit produit;  
    private int quantite;  
    // Getters et Setters  
    // Constructeurs  
}
```

Panier

```
package net.youssfi.eboutique.web.model;import java.util.*;
import net.youssfi.eboutique.entities.Produit;
public class Panier {
    private Map<Long, ArticlePanier> articles=new HashMap<Long, ArticlePanier>();
    public void ajouterArticle(Produit p,int quantite){
        ArticlePanier art=articles.get(p.getIdProduit());
        if(art!=null) art.setQuantite(art.getQuantite()+quantite);
        else articles.put(p.getIdProduit(), new ArticlePanier(p, quantite));
    }
    public Collection<ArticlePanier> getArticles(){
        return articles.values();
    }
    public double getTotal(){
        double total=0;
        Collection<ArticlePanier> items=getArticles();
        for(ArticlePanier art:items){
            total+=art.getQuantite()*art.getProduit().getPrix();
        }
        return total;
    }
}
```

Panier

```
public void deleteItem(Long idProduit){
    articles.remove(idProduit);
}

public int getSize(){
    int nb=0;
    Collection<ArticlePanier> items=getArticles();
    for(ArticlePanier art:items){
        nb+=art.getQuantite();
    }
    return nb;
}

}
```



Contrôleurs

CatagoriesManagerController

```
package net.youssfi.eboutique.web.controllers;
import java.awt.image.BufferedImage;
import java.io.*;import javax.imageio.ImageIO;
import javax.servlet.http.*;
import javax.validation.Valid;
import org.apache.commons.io.IOUtils;
import net.youssfi.eboutique.entities.*;
import net.youssfi.eboutique.metier.*;
import org.slf4j.*;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.MediaType;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.multipart.*;
import org.springframework.web.servlet.HandlerExceptionResolver;
import org.springframework.web.servlet.ModelAndView;
```

CatagoriesManagerController

```
@Controller
@RequestMapping(value="/categories")
@SessionAttributes("cat")
public class CatagoriesManagerController implements
    HandlerExceptionResolver {
    Logger logger=LoggerFactory.getLogger(this.getClass());
    @Autowired
    private ICategoriesManager metier;
    @RequestMapping(value="/index")
    public String categories(Model model){
        model.addAttribute("categories",metier.listCategories() );
        model.addAttribute("categorie", new Categorie());
        return "CategoriesView";
    }
}
```


CatagoriesManagerController

```
@RequestMapping(value="/enregistrer")
public String enregistrer( @Valid Categorie c, BindingResult bindingResult,
    Model model, MultipartFile file) throws Exception{
    if(bindingResult.hasErrors()){
        model.addAttribute("categories",metier.listCategories() );
        return"CategoriesView";
    }
    if(!file.isEmpty()){
        BufferedImage bi=ImageIO.read(file.getInputStream());
        c.setNomPhoto(file.getOriginalFilename()); c.setPhoto(file.getBytes());
    }
    else{
        Categorie cat=(Categorie) model.asMap().get("cat");c.setPhoto(cat.getPhoto());
    }
    if(c.getIdCategorie()==null) metier.ajouterCategorie(c);
    else metier.modifierCategorie(c);
    model.addAttribute("categories",metier.listCategories() );
    model.addAttribute("categorie", new Categorie());
    return "CategoriesView";
}
```

CatagoriesManagerController

```
@ResponseBody
@RequestMapping(value="/photoCat",produces=MediaType.IMAGE_JPEG_VALUE)
public byte[] photo(@RequestParam(value="idCat")Long idCat)throws IOException{
    Categorie c=metier.getCategorie(idCat);
    return IOUtils.toByteArray(new ByteArrayInputStream(c.getPhoto()));
}

@ExceptionHandler(MaxUploadSizeExceededException.class)
public ModelAndView resolveException(HttpServletRequest req,
    HttpServletResponse arg1, Object arg2, Exception exception) {
    logger.error("Request: " + req.getRequestURL() + " raised " + exception);
    exception.printStackTrace();

    ModelAndView mav = new ModelAndView();
    mav.addObject("errors", exception.getMessage());
    mav.addObject("url", req.getRequestURL());
    mav.setViewName("CategoriesView");
    mav.addObject("categories",metier.listCategories() );
    mav.addObject("categorie", new Categorie());
    return mav;
}
```

CatagoriesManagerController

```
@RequestMapping("/suppCat")
public String deleteCat(@RequestParam(value="idCat") Long idCat,Model mdeol){
    metier.supprimerCategrorie(idCat);
    mdeol.addAttribute("categories",metier.listCategories() );
    mdeol.addAttribute("categorie", new Categorie());
    return "CategoriesView";
}

@RequestMapping("/editCat")
public String editCat(@RequestParam(value="idCat") Long idCat,Model model){
    Categorie c=metier.getCategorie(idCat);
    model.addAttribute("categories",metier.listCategories() );
    model.addAttribute("categorie",c);
    model.addAttribute("cat", c);
    return "CategoriesView";
}
}
```

ProduitsManagerController

```
package net.youssfi.eboutique.web.controllers;
import java.io.File;
import java.util.List;
import javax.validation.Valid;
import net.youssfi.eboutique.entities.Categorie;
import net.youssfi.eboutique.entities.Produit;
import net.youssfi.eboutique.metier.IProduitsManager;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.ResponseBody;
import org.springframework.web.multipart.MultipartFile;
```

ProduitsManagerController

```
@Controller
@RequestMapping("/produits")
public class ProduitsManagerController {
    @Autowired
    private IProduitsManager metier;
    @RequestMapping(value="/index")
    public String index(Model model){
        model.addAttribute("produit", new Produit());
        model.addAttribute("categories", metier.listCategories());
        model.addAttribute("produits", metier.listproduits());
        return "ProduitsView";
    }
}
```

ProduitsManagerController

```
@RequestMapping(value="/saveProduit")
public String enregistrer(@Valid Produit p, BindingResult bindingResult,
    MultipartFile file,
    Model model) throws Exception{
    if(bindingResult.hasErrors()) {
        model.addAttribute("produits", metier.listproduits());
        return "ProduitsView« ; }
    if(!file.isEmpty()){ p.setPhoto(file.getOriginalFilename()); }
    if(p.getIdProduit()==null){
        metier.ajouterProduit(p, p.getCategorie().getIdCategorie());
    } else{ metier.modifierProduit(p); }
    if(!file.isEmpty()){
        String path=System.getProperty("java.io.tmpdir")+"/"+p.getIdProduit();
        file.transferTo(new File(path));
    }
    model.addAttribute("produit", new Produit());
    model.addAttribute("produits", metier.listproduits());
    return "ProduitsView";
}
```

ProduitsManagerController

```
@ModelAttribute("categories")
public List<Categorie> listCategories(){
return metier.listCategories();
}

@RequestMapping(value="/deleteProd")
public String deleteProduit(@RequestParam("idP")Long idP,Model model){
metier.supprimerProduit(idP);
model.addAttribute("produits", metier.listproduits());
model.addAttribute("produit",new Produit());
return "ProduitsView";
}

@RequestMapping(value="/editProd")
public String editProduit(@RequestParam("idP")Long idP,Model model){
Produit p=metier.getProduit(idP);
model.addAttribute("produit",p);
model.addAttribute("produits", metier.listproduits());
return "ProduitsView";
}
}
```

LoginController

```
package net.youssfi.eboutique.web.controllers;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
@Controller
public class LoginController {
    @RequestMapping("/login")
    public String login(){
        return "login";
    }
}
```


EBoutiqueController

```
package net.youssfi.eboutique.web.controllers;
import java.io.FileInputStream;
import net.youssfi.eboutique.entities.Produit;
import net.youssfi.eboutique.metier.IUserCatalogue;
import net.youssfi.eboutique.web.model.Panier;
import org.apache.commons.io.IOUtils;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.MediaType;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.ResponseBody;
import org.springframework.web.bind.annotation.SessionAttributes;
```

EBoutiqueController

```
@Controller
@SessionAttributes("panier")
public class EBoutiqueController {
    @Autowired
    private IUserCatalogue metier;
    @RequestMapping("/index")
    public String index(Model model){
        if(model.asMap().get("panier")==null){
            model.addAttribute("panier", new Panier());
        }
        model.addAttribute("categories", metier.listCategories());
        model.addAttribute("produits", metier.produitsSelectionnes());
        return "index";
    }
}
```

EBoutiqueController

```
@RequestMapping("/produitsParCat")
public String produitsParCat(@RequestParam Long idCat, Model
model){
    model.addAttribute("categories", metier.listCategories());
    model.addAttribute("produits",
    metier.produitsParCategorie(idCat));
    return "index";
}

@RequestMapping("/chercherProduits")
public String chercherProduits(@RequestParam String mc, Model
model){
    model.addAttribute("mc", mc);
    model.addAttribute("categories", metier.listCategories());
    model.addAttribute("produits", metier.produitsParMotCle(mc));
    return "index";
}
```

EBoutiqueController

```
@RequestMapping(value="/photoProduit",produces=MediaType.IMAGE_JPEG_
    VALUE)
@ResponseBody
public byte[] photoProd(@RequestParam("idP")Long idP) throws
    Exception{
    Produit p=metier.getProduit(idP);
    String
        path=System.getProperty("java.io.tmpdir")+"/"+p.getIdProduit();
    return IOUtils.toByteArray(new FileInputStream(path));
}
```

EBoutiqueController

```
@RequestMapping("/ajouterAuPanier")  
public String chercherProduits(@RequestParam Long  
    idProduit,@RequestParam int quantite,Model model){  
    Panier panier=null;  
    if(model.asMap().get("panier")==null){  
        panier=new Panier();  
        model.addAttribute("panier", panier);  
    }  
    else  
        panier=(Panier) model.asMap().get("panier");  
    panier.ajouterArticle(metier.getProduit(idProduit), quantite);  
    model.addAttribute("categories", metier.listCategories());  
    model.addAttribute("produits", metier.produitsSelectionnees());  
    return "index";  
    }  
}
```



Les Vues

CategoriesView.jsp

```
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@taglib uri="http://www.springframework.org/tags/form" prefix="f"%>
<f:form modelAttribute="categorie" action="enregistrer"
  enctype="multipart/form-data">
  <div class="error">
    ${errors}
  </div>
  <table>
    <tr>
      <td>ID Catégorie:</td>
      <td><f:input type="hidden" path="idCategorie"/>
        ${categorie.idCategorie}
      </td>
    </tr>
    <tr>
      <td>Nom Catégorie</td>
      <td><f:input path="nomCategorie"/></td>
      <td><f:errors path="nomCategorie" cssClass="errors"></f:errors></td>
    </tr>
```

CategoriesView.jsp

```
<tr>
  <td>Description</td>
  <td><f:textarea path="description" cols="50" rows="5"/></td>
  <td><f:errors path="nomCategorie" cssClass="errors"></f:errors></td>
</tr>
<tr>
  <td>Photo</td>
  <td>
    <c:if test="${categorie.idCategorie!=null}">
      
    </c:if>
    <input name="file" type="file"/></td>
  <td><f:errors path="photo" cssClass="errors"></f:errors></td>
</tr>
<tr>
  <td><input type="submit" value="Enregistrer"></td>
</tr>
</table>
</f:form>
```


CategoriesView.jsp

```
<table class="table1">
  <tr>
    <th>ID</th><th>NOM CAT</th> <th>Description</th><th>Photo</th>
  </tr>
  <c:forEach items="${categories}" var="cat">
    <tr>
      <td>${cat.idCategorie }</td>
      <td>${cat.nomCategorie}</td>
      <td>${cat.description}</td>
      <td></td>
      <td><a href="suppCat?idCat=${cat.idCategorie }">Supprimer</a></td>
      <td><a href="editCat?idCat=${cat.idCategorie }">Edit</a></td>
    </tr>
  </c:forEach>
</table>
```

ProduitsView.jsp

```
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@taglib uri="http://www.springframework.org/tags/form" prefix="f"%>
<div>
  <a href="<c:url value="/j_spring_security_logout" />" > Logout</a>
  <form action="chercherProduits" action="chercherProduits">
    <table>
      <tr>
        <td><input type="text" name="mc" value="${mc}"></td>
        <td><input type="submit" value="Chercher"></td>
      </tr>
    </table>
  </form>
</div>
```

ProduitsView.jsp

```
<div>
  <f:form modelAttribute="produit" action="saveProduit"
    enctype="multipart/form-data">
    <table>
      <tr>
        <td>ID Produit</td>
        <td><f:input type="hidden" path="idProduit"/>${produit.idProduit}</td>
        <td><f:errors path="idProduit" cssClass="errors"></f:errors></td>
      </tr>
      <tr>
        <td>Categorie</td>
        <td><f:select
          path="categorie.idCategorie"
          items="${categories}"
          itemLabel="nomCategorie"
          itemValue="idCategorie" /></td>
        <td><f:errors path="categorie.idCategorie"
          cssClass="errors"></f:errors></td>
      </tr>
    </table>
  </f:form>
</div>
```

ProduitsView.jsp

```
<tr>
  <td>Désignation</td>
  <td><f:input path="designation"/></td>
  <td><f:errors path="designation" cssClass="errors"></f:errors></td>
</tr>
<tr>
  <td>Description</td>
  <td><f:input path="description"/></td>
  <td><f:errors path="description" cssClass="errors"></f:errors></td>
</tr>
<tr>
  <td>Prix</td>
  <td><f:input path="prix"/></td>
  <td><f:errors path="prix" cssClass="errors"></f:errors></td>
</tr>
```

ProduitsView.jsp

```
<tr>
  <td>Sélectionné</td>
  <td><f:checkbox path="selectionne"/></td>
  <td><f:errors path="selectionne" cssClass="errors"></f:errors></td>
</tr>
<tr>
  <td>Quantité</td>
  <td><f:input path="quantite"/></td>
  <td><f:errors path="quantite" cssClass="errors"></f:errors></td>
</tr>
```

ProduitsView.jsp

```
<tr>
    <td>Photo</td>
    <td>
        <c:if test="${produit.idProduit!=null}">
            <f:input type="hidden" path="photo"/>
            
        </c:if>
        <input type="file" name="file"></td>
    <td>${errors }</td>
</tr>
<tr>
    <td><input type="submit" value="enregistrer"></td>
</tr>
</table>
</f:form>
</div>
```

ProduitsView.jsp

```
<div>
  <table class="table1">
    <tr>
      <th>ID</th><th>Designation</th><th>Prix</th><th>Sélectionnée</th>
      <th>Quatité</th><th>Photo</th>
    </tr>
    <c:forEach items="${ produits}" var="p">
      <tr>
        <td>${p.idProduit }</td> <td>${p.designation }</td>
        <td>${p.prix}</td> <td>${p.selectionne}</td>
        <td>${p.quantite}</td>
        <td></td>
        <td><a href="deleteProd?idP=${p.idProduit }">Supp</a></td>
        <td><a href="editProd?idP=${p.idProduit }">Edit</a></td>
      </tr>
    </c:forEach>
  </table>
</div>
```

login.jsp

```
<!DOCTYPE html>
<html><head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Login</title>
</head>
<body>
  <div class="cadre">
    <form action="j_spring_security_check" method="post">
      <table>
        <tr>
          <td>Login</td> <td><input type="text" name="j_username"></td>
        </tr>
        <tr>
          <td>Pass word</td> <td><input type="password" name="j_password"></td>
        </tr>
        <tr> <td><input type="submit" value="Login"></td> </tr>
      </table>
    </form>
  </div>
</body>
</html>
```


index.jsp

```
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<div id="panier" style="display: none">
  <c:if test="${panier.size!=0}">
    <table>
      <tr>
        <th>ID</th><th>Désignation</th><th>Prix</th><th>Quantité</th><th>Montant</th>
      </tr>
      <c:forEach items="${panier.articles}" var="art">
        <tr>
          <td>${art.produit.idProduit}</td> <td>${art.produit.designation}</td>
          <td>${art.produit.prix}</td><td>${art.quantite}</td>
          <td>${art.quantite*art.produit.prix}</td>
        </tr>
      </c:forEach>
      <tr>
        <td colspan="4">Total</td>
        <td>${panier.total}</td>
      </tr>
    </table>
  </c:if>
</div>
```

index.jsp

```
<div id="catalogueProduits">
  <c:forEach items="${produits}" var="p">
    <div class="ficheProduit">
      <table>
        <tr><td colspan="2"></td> </tr>
        <tr> <td>Désignation :</td><td>${p.designation }</td> </tr>
        <tr> <td>Prix :</td><td>${p.prix}</td> </tr>
        <tr> <td>Stock:</td><td>${p.quantite}</td> </tr>
        <tr> <td>${p.description }</td></tr>
        <tr> <td colspan="2">
          <form action="ajouterAuPanier">
            <input type="hidden" value="${p.idProduit}" name="idProduit">
            <input type="text" value="1" name="quantite">
            <input type="submit" value="Ajouter au panier">
          </form>
        <td>
      </tr>
    </table>
  </div>
</c:forEach>
</div>
```