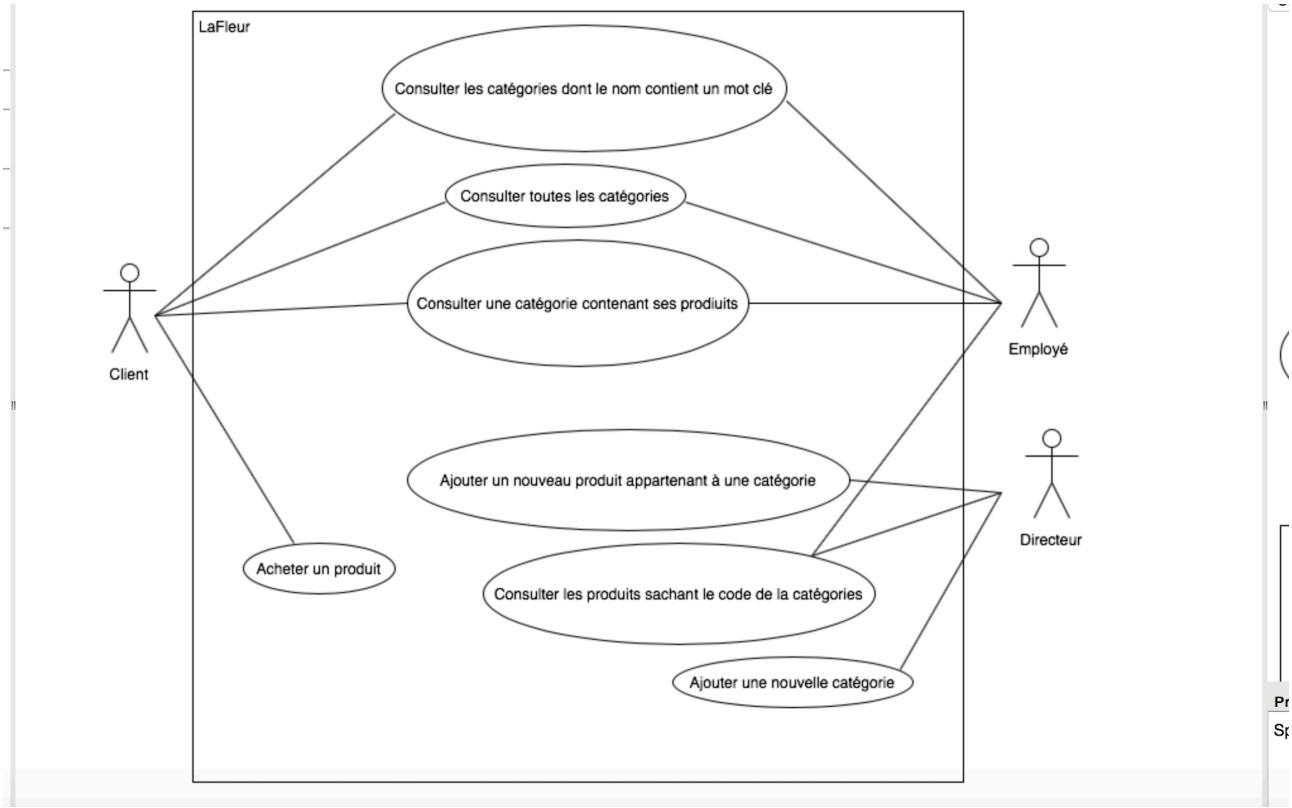
TP4-Ravao Yoan

Diagramme de cas d'utilisation:



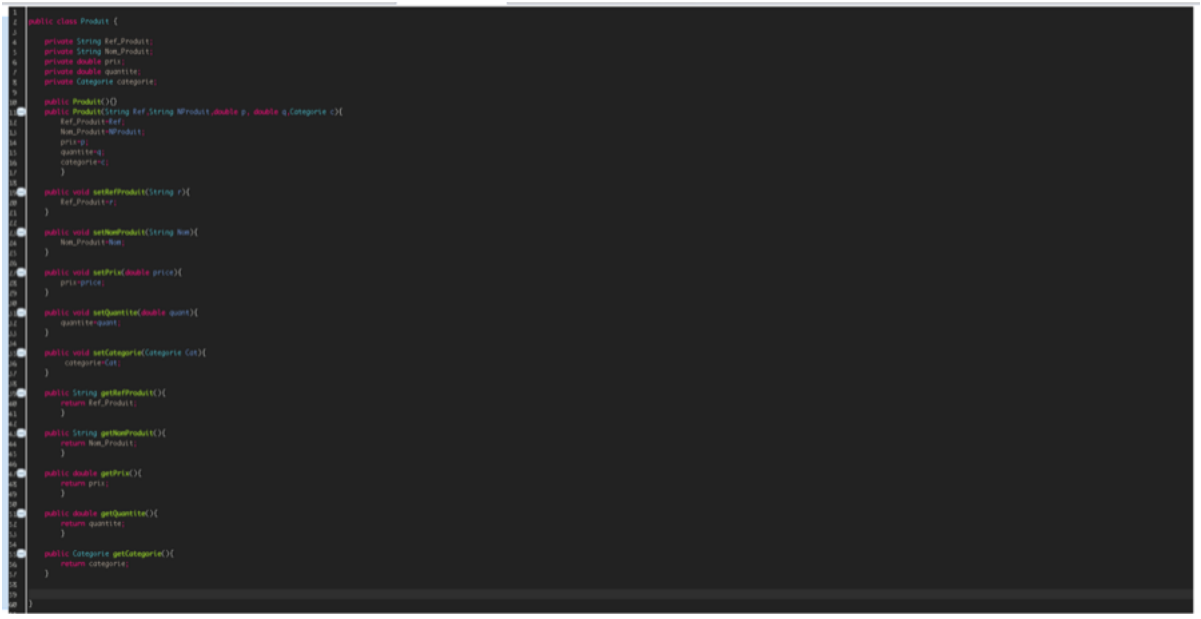
je commence a crée le MLDR:

Après le diagramme de cas d'utilisation je crée les classe persistante Catégorie et Produit.

Catégorie: Je crée le Constructeur ainsi que les getters et les setters

```java
public class Categorie {
    private int idCat;
    private String nomCat;
    private List <Produit> produit=new ArrayList<Produit>();


    // Constructeur
    public Categorie(){}
    public Categorie(int Numcat,String nom){
        idCat=Numcat;
        nomCat=nom;
        }


    //setters

    public void  setNomCat(String Ncat){
        nomCat=Ncat;
    }

    public void setIdCat(int id){
        idCat=id;
    }



    //getters

    public int getProduitparCat(){
        return idCat;
    }

    public String getNomCat(){
        return nomCat;
    }



}
```

Produit: Je fais de même avec produit



```java
public class Produit {

    private String Ref_Produit;
    private String Nom_Produit;
    private double prix;
    private double quantite;
    private Categorie categorie;

    public Produit(){}
    public Produit(String Ref,String NProduit,double p, double q,Categorie c){
        Ref_Produit=Ref;
        Nom_Produit=NProduit;
        prix=p;
        quantite=q;
        categorie=c;
        }

    public void setRefProduit(String r){
        Ref_Produit=r;
    }

    public void setNomProduit(String Nom){
        Nom_Produit=Nom;
    }

    public void setPrix(double price){
        prix=price;
    }

    public void setQuantite(double quant){
        quantite=quant;
    }

    public void setCategorie(Categorie Cat){
        categorie=Cat;
    }

    public String getRefProduit(){
        return Ref_Produit;
        }

    public String getNomProduit(){
        return Nom_Produit;
        }

    public double getPrix(){
        return prix;
        }

    public double getQuantite(){
        return quantite;
        }

    public Categorie getCategorie(){
        return categorie;
    }

}
```

Création de la classe **SingletonConnection** qui permettra de retourner une Connection vers la Base de données:

```java
import java.sql.*;


public class SingletonConnection {
    private static Connection connection;

    static{
        try{
            Class.forName("com.mysql.jdoc.Driver");
            connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/DB_","root","");

        }catch (Exception e) {
            e.printStackTrace();
        }
    }
    public static Connection getConnetion(){
        return connection;
    }
}
```

On a ensuite **L'interface IMétier:**

```java
import java.util.*;

public interface Imetier {

      public void addCategorie (Categorie c);
      public void addProduit(Produit p, int idCat);
      public List <Produit> getProduitparMC(String MC);
      public List <Produit> getProduitparCat(int idCat);
      public List <Produit> getAllCatgorie();
      public Categorie getCategorie(int idCat);
}
```

Enfin pour finir on crée le main qui permettra de Tester l'ensemble de la couche métier.

```java
import java.util.*;


public class Test {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        for (Categorie c:cats){
            System.out.println(c.getNomCat());
            System.out.println(c.ToString());
        }

        System.out.println("--------------------");
        List<Produit> prods2=metier.getproduitParMC("D");

        for(Produit p:prods2){
            System.out.println(p.getNomProduit());
            System.out.println(p.getPrix());
            System.out.println(p.getCategorie().getNomCat());
        }
    }

}

```