

Gestion Projet JAVA

Hamza EL GHAZOUANI

Les objectif du Projet :

- Réaliser le diagramme de classe a de quoi ce mcd
 - Réaliser l'architecture d'un projet de développement java permettant de réaliser un application traitant ce projet
 - Réaliser une classe permettant la connections a la base de données est respecter mcd
 - Réaliser une interface graphique contenant un menu permettant d'accéder, gérer les employer , les service , entreprise
 - Pour les trois interface 's ,s ,m' mettre un Jtable permettant d'afficher les données de chacune du table un btn permettant d'ouvrir un nouveau forme pour ajoute un new enregistrement + Un btn modifier ouvrir le même forme est modifier l'enregistrement sélectionner
-

MCD

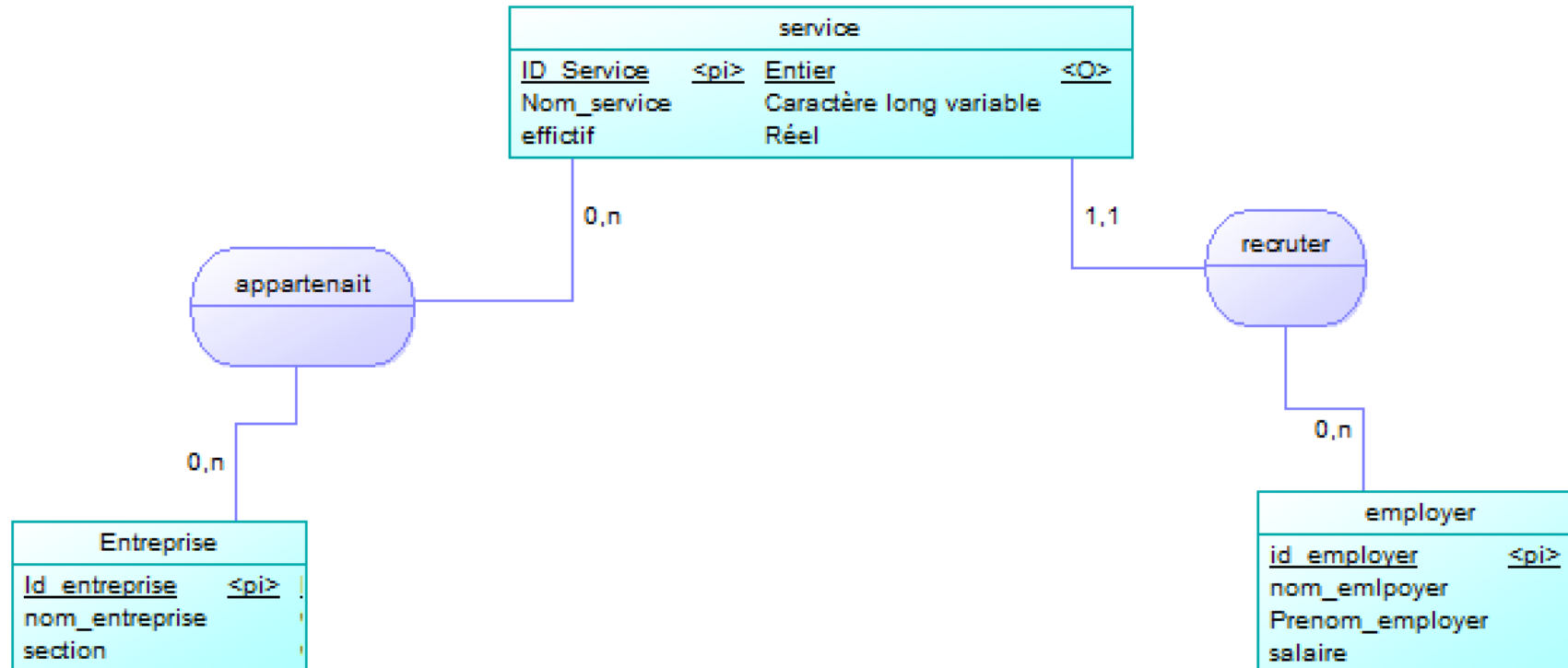
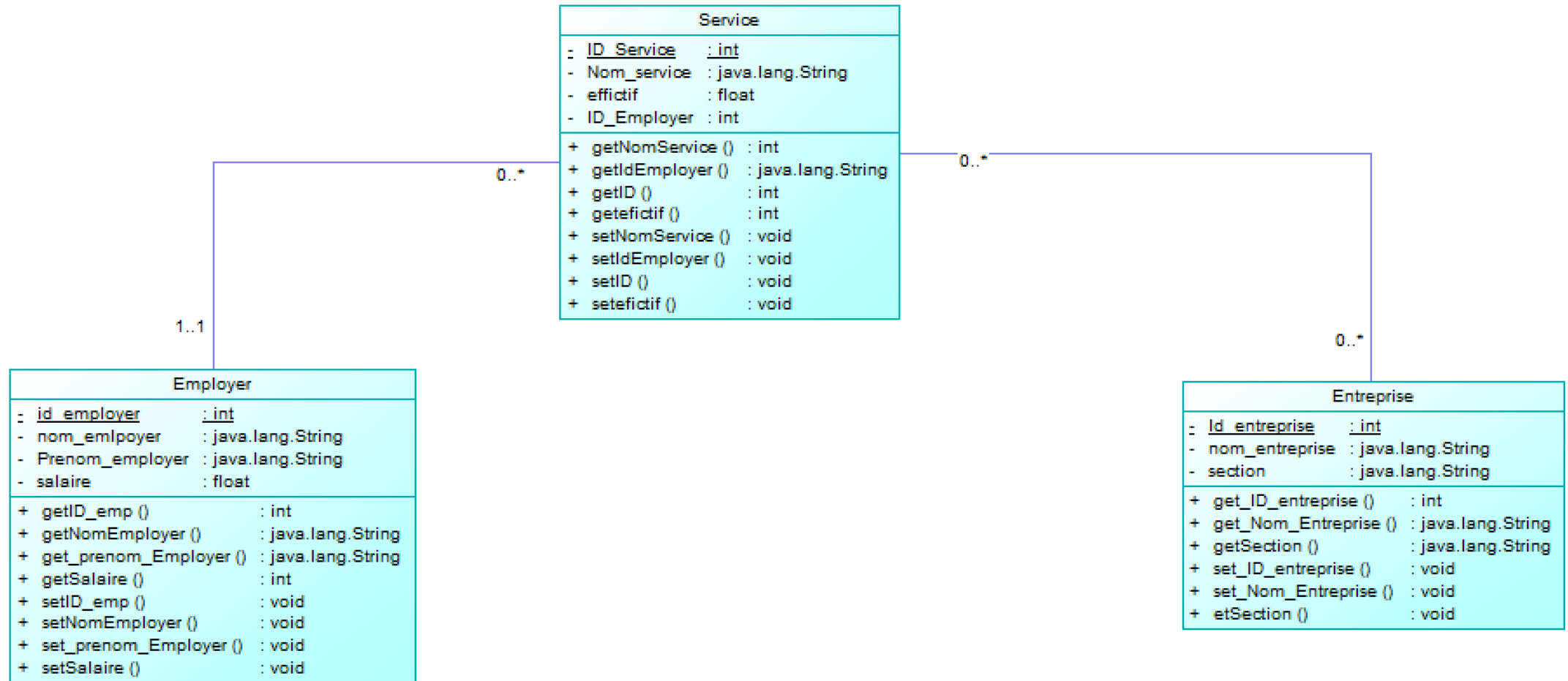
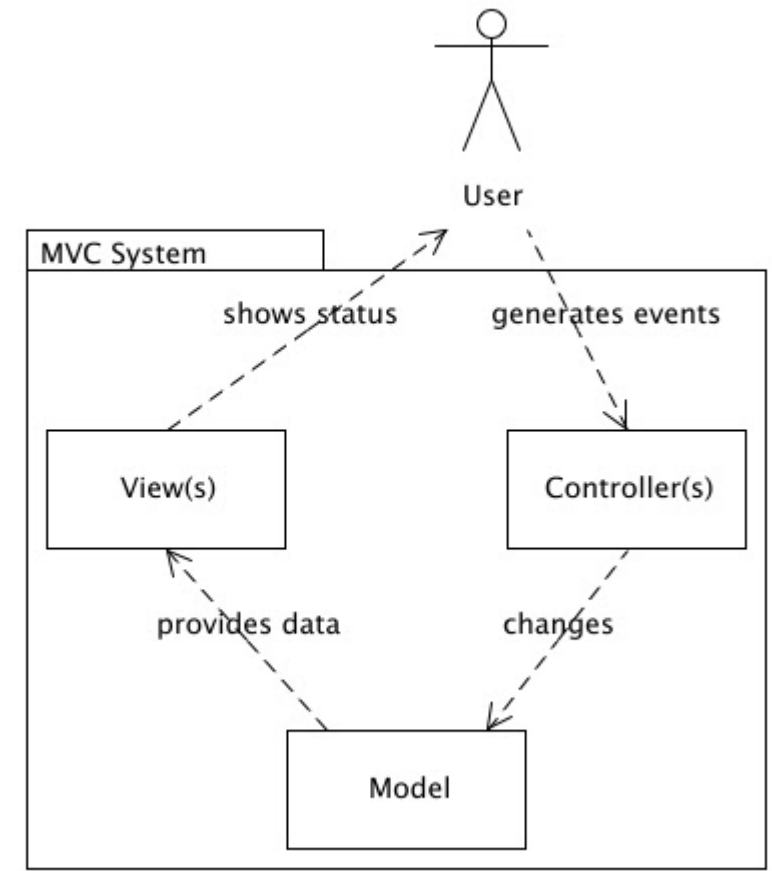
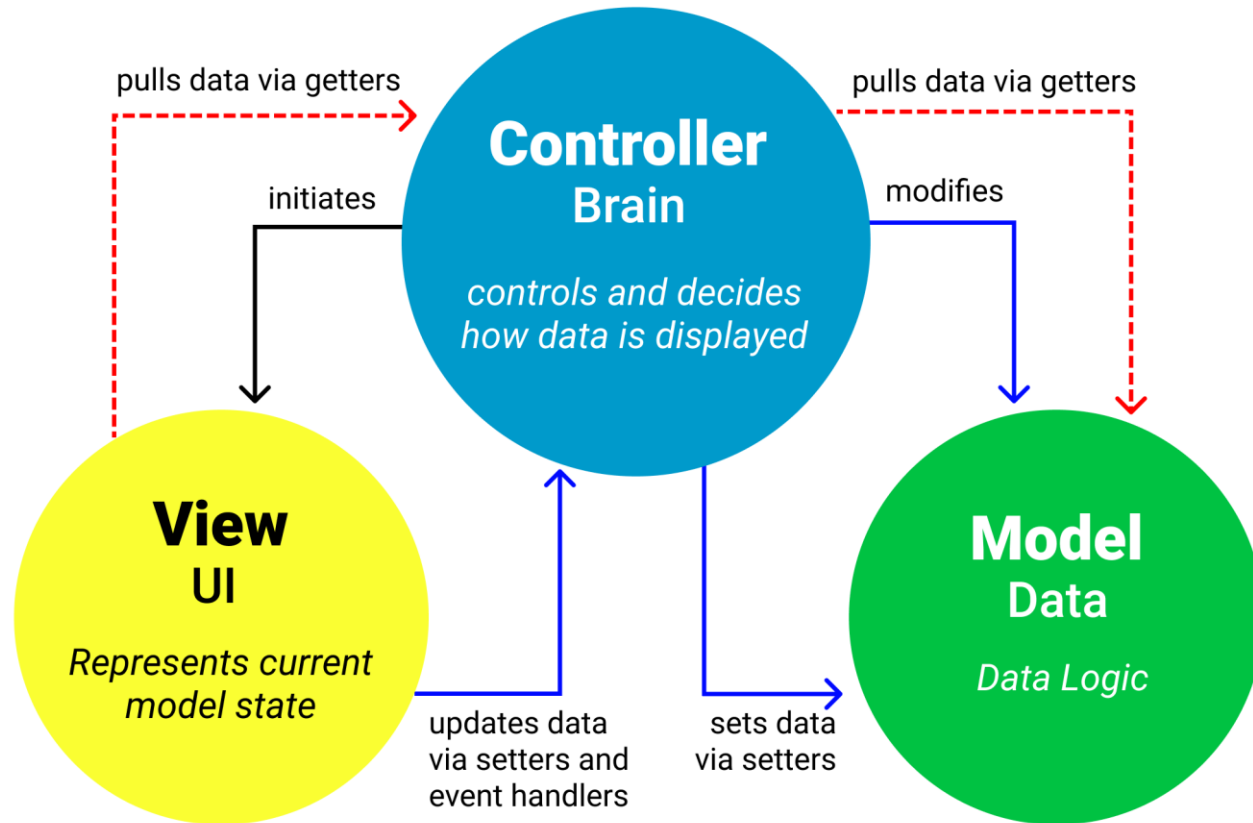


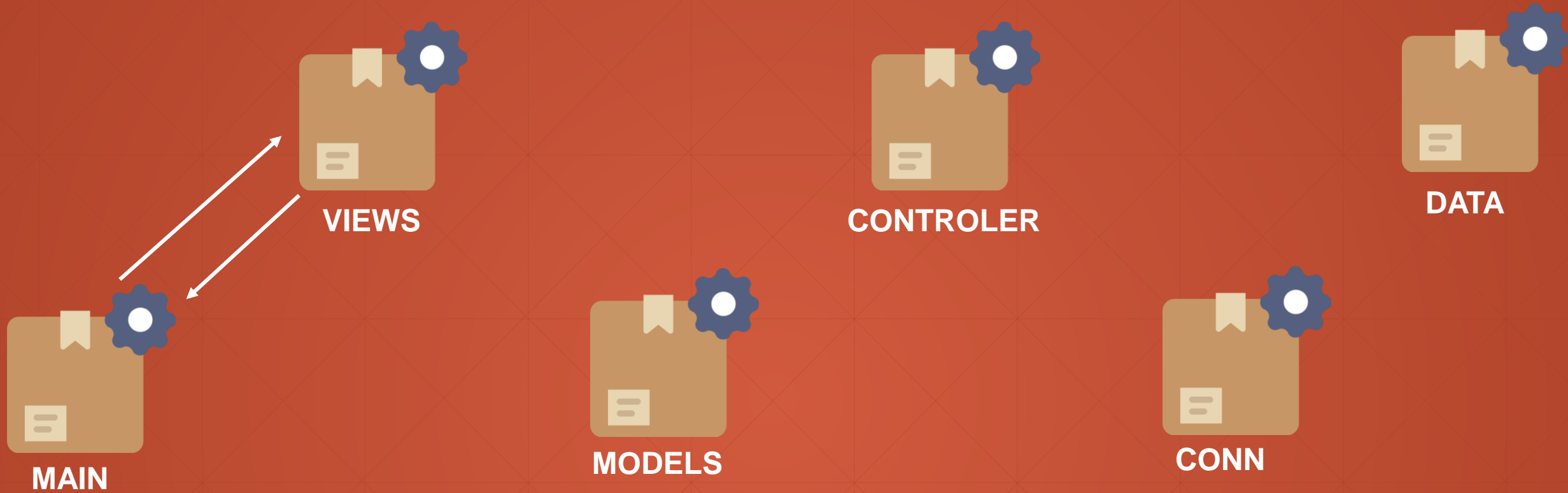
Diagramme de Classe



Basé sur L'Architecture :

MVC Architecture Pattern





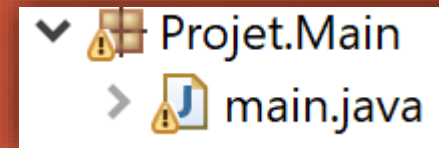
EXPLICATION D'ARCHETECTUTRE

PACKAGE



Main Projet

Contiens une creation
d'objet « ViewMain »














```
package Projet.Main;

import javax.swing.UIManager;

public class main {

    @SuppressWarnings("unused")
    public static void main(String[] args) throws UnsupportedOperationException {
        UIManager.setLookAndFeel(new NimbusLookAndFeel());
        //creation d'un constructeur de type view
        //qui permet de d'ouvrir la fenetre principale ViewMain
        ViewMain man = new ViewMain() ;
    }
}
```

- ▼  Views
 -  TableEmployer.java
 -  TableEntreprise.java
 -  TableService.java
 -  ViewAjouterEmployer.java
 -  ViewAjouterEntreprise.java
 -  ViewAjouterService.java
 -  ViewMain.java
 -  ViewUpdateEmployer.java
 -  ViewUpdateEntremprise.java
 -  ViewUpdateService.java

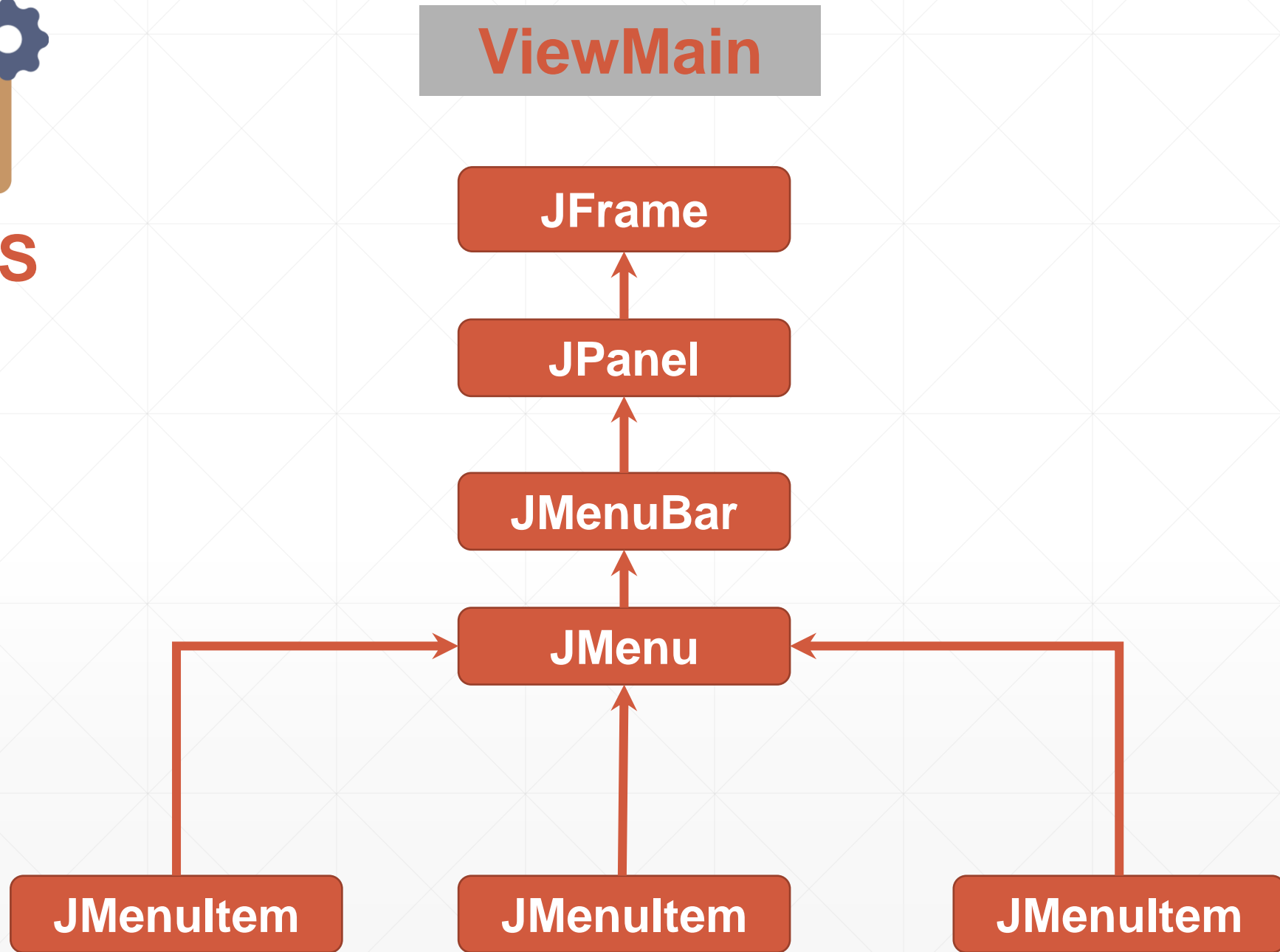


VIEWS

Contiens une création
La définition de tout
les vues



VIEWS





VIEWS

VIEWFORM *

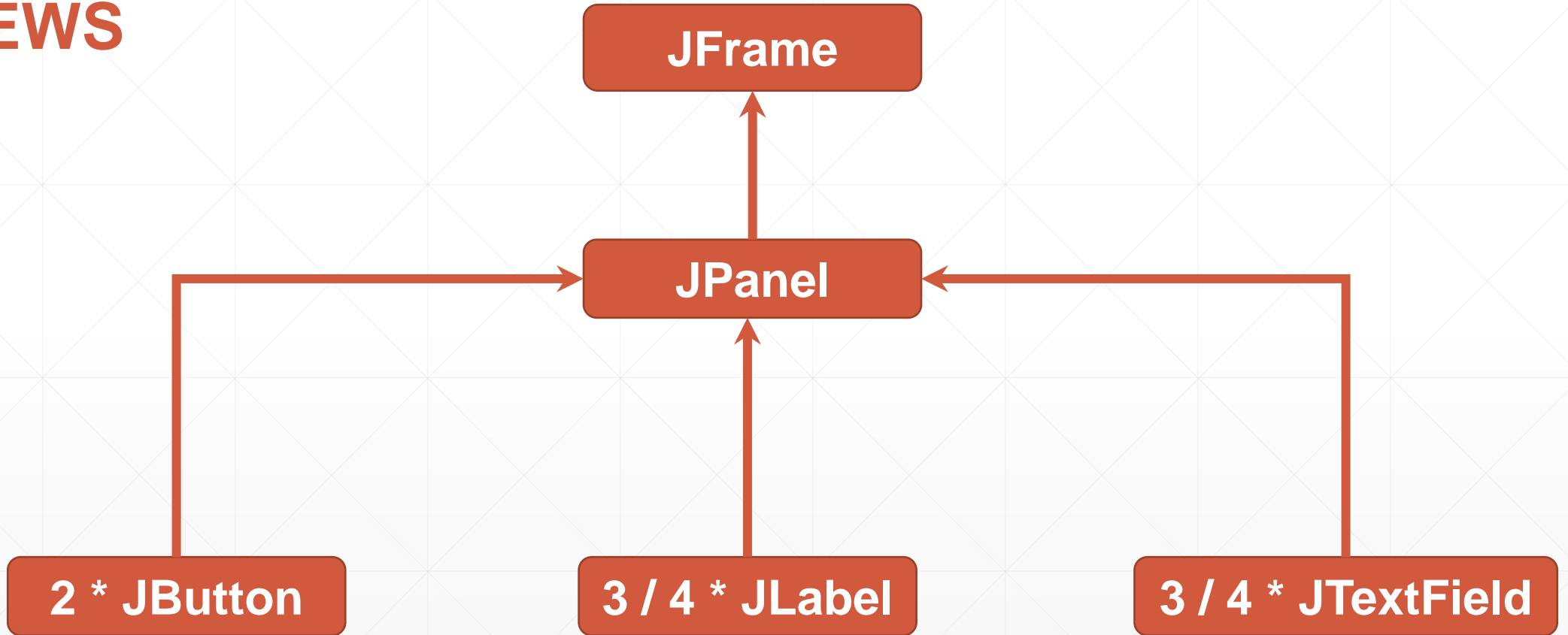
JFrame

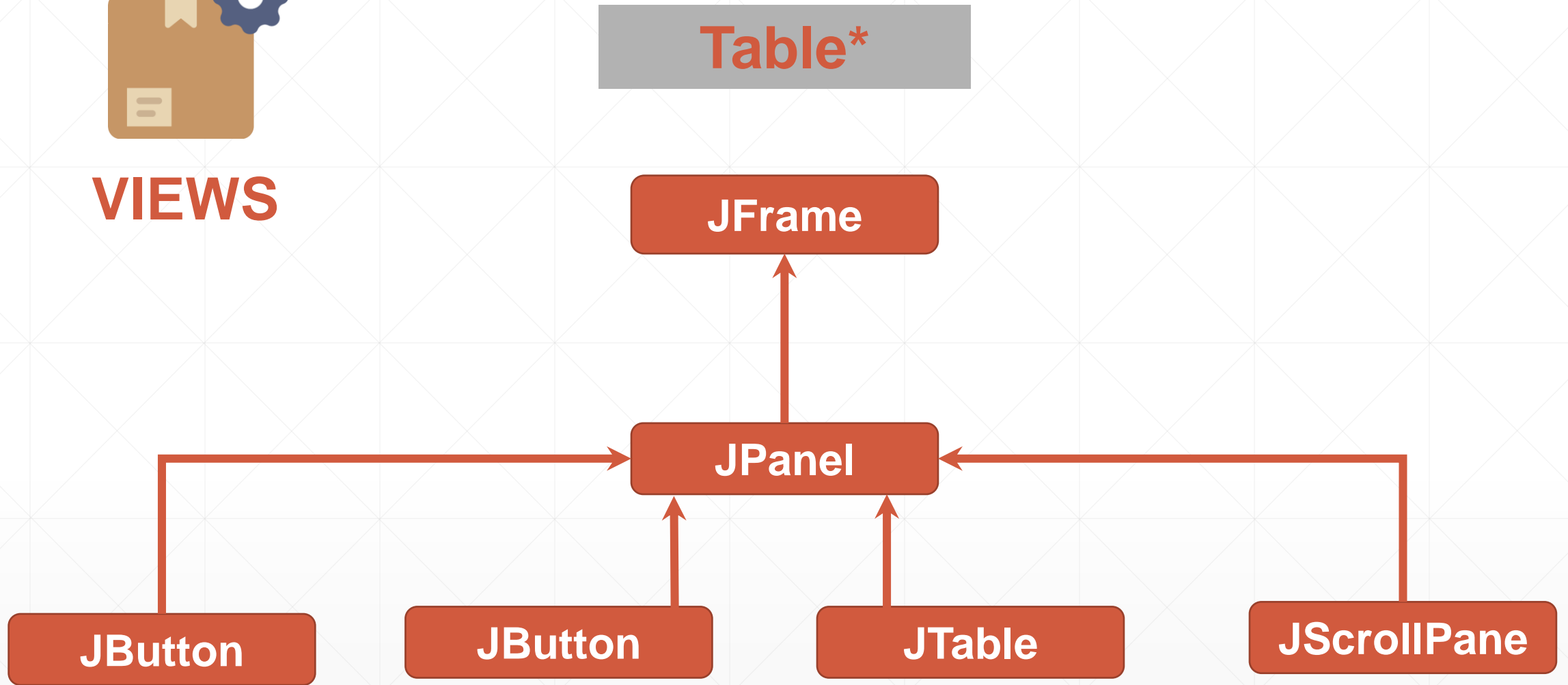
JPanel












2 * JButton

3 / 4 * JLabel

3 / 4 * JTextField









- ▼  Contrôler
 -  AddModifierEmp.java
 -  AddModifierEnt.java
 -  AddModifierServ.java
 -  ContrôlerFormEmp.java
 -  ContrôlerFormEntr.java
 -  ContrôlerFormServ.java
 -  ContrôlerTables.java
 -  UpdateModifieremp.java
 -  UpdateModifierEnt.java
 -  UpdateModifiervers.java



CONTROLLERS

Contiens Tout les
action du VIEW



- ▼  Models
 -  Employer.java
 -  Entreprise.java
 -  Service.java

MODELS



Contiens les classes
De transmission des
données







Conn

Contiens L'ouverture
et la fermeture de la
connections avec
notre base de
données

```
public class connection {  
    public Connection con() {  
        try  
        {  
            //étape 1: charger la classe de driver  
            Class.forName("com.mysql.jdbc.Driver");  
  
            //étape 2: créer l'objet de connexion  
            Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/gestion", "root", "");  
            System.out.println("valider");  
  
            return conn;  
        }  
        catch(Exception e){  
            System.out.println("Exeption : "+e);  
            return null;  
        }  
    }  
  
    public Boolean closecon(Connection conn) {  
        try  
        {  
            System.out.println("close");  
            conn.close();  
            return true;  
        }  
        catch(Exception e){  
            System.out.println("Exeption : "+e);  
            return null;  
        }  
    }  
}
```

▼  Conn
 >  connection.java

- ▼  Data
 -  DataEmployer.java
 -  DataEntreprise.java
 -  DataService.java



Data

Contiens tout les
requête SQL