



APPLICATION ADMINISTRATIVE

DOCUMENTATION TECHNIQUE

Projet E4

Application Java / Java FX

Cheik-Siramakan Keita

cheiksiramakankeita@gmail.com



TABLE DES MATIERES

2 Contexte et cahier des charges	2
3 Outils utilisÉs	3
4 Base de donnÉes	4
MCD	4
Tables	4
5 Objets	5
6 DAO	6
7 Controller	7
8 Vue	8
9 Gestion des erreurs	9

2 CONTEXTE ET CAHIER DES CHARGES

Une CFA spécialisé dans les métiers du bâtiment a fait appel à la société ZenMedia, une SSII afin de lui développer un logiciel de gestion pour son organisation.

Cette application devra permettre aux employés de gérer les différents utilisateurs facilement.

Les utilisateurs de cette application devront pouvoir naviguer dans les menus simplement et sans tracas.

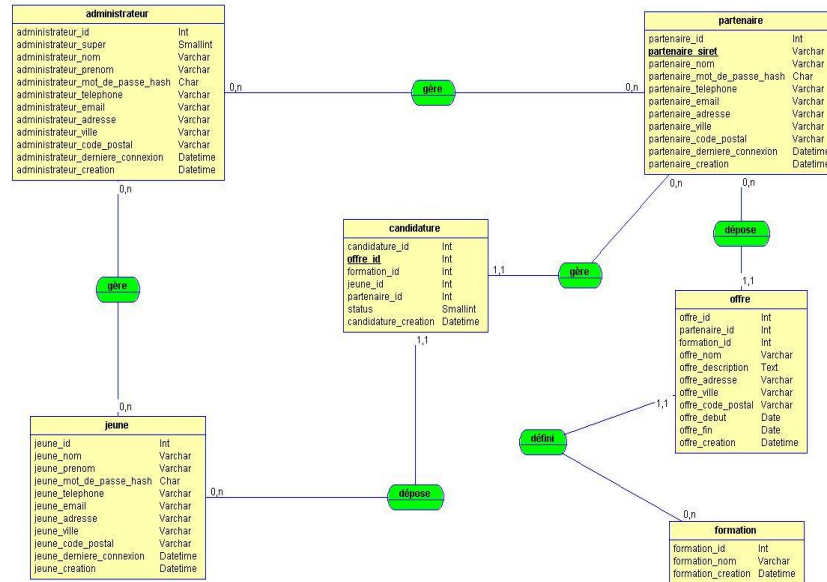
3 OUTILS UTILISÉS

Les outils utilisés sont Eclipse, Scenebuilder, MySQL et Git.



4 BASE DE DONNÉES






MCD



TABLES

 administrateur		 Parcourir	 Structure	 Rechercher	 Insérer	 Vider	 Supprimer	2	InnoDB	utf8_general_ci	32,0	kio	-
 formation		 Parcourir	 Structure	 Rechercher	 Insérer	 Vider	 Supprimer	3	InnoDB	utf8_general_ci	32,0	kio	-
 jeune		 Parcourir	 Structure	 Rechercher	 Insérer	 Vider	 Supprimer	2	InnoDB	utf8_general_ci	32,0	kio	-
 offre		 Parcourir	 Structure	 Rechercher	 Insérer	 Vider	 Supprimer	3	InnoDB	utf8_general_ci	64,0	kio	-
 partenaire		 Parcourir	 Structure	 Rechercher	 Insérer	 Vider	 Supprimer	2	InnoDB	utf8_general_ci	32,0	kio	-
5 tables	Somme							12	MyISAM	utf8_general_ci	192,0	kio	0

5 OBJETS

- >  Administrateur.java
- >  Formation.java
- >  Jeune.java
- >  Offre.java
- >  Partenaire.java

Ces classes ont toujours la même structure avec les getters et les setters et le constructeur.

```
package classe;

import javafx.beans.property.IntegerProperty;

public class Administrateur
{
    private IntegerProperty administrateur_id;
    private StringProperty administrateur_nom;
    private StringProperty administrateur_prenom;
    private StringProperty administrateur_identifiant;
    private StringProperty administrateur_mot_de_passe;
    private StringProperty administrateur_email;
    private StringProperty administrateur_telephone;
    private StringProperty administrateur_adresse;
    private StringProperty administrateur_ville;
    private StringProperty administrateur_code_postal;
    private StringProperty administrateur_derniere_connexion;
    private StringProperty administrateur_date_ajout;






    public Administrateur()
    {
        this.administrateur_id = new SimpleIntegerProperty();
        this.administrateur_nom = new SimpleStringProperty();
        this.administrateur_prenom = new SimpleStringProperty();
        this.administrateur_identifiant = new SimpleStringProperty();
        this.administrateur_mot_de_passe = new SimpleStringProperty();
        this.administrateur_email = new SimpleStringProperty();
        this.administrateur_telephone = new SimpleStringProperty();
        this.administrateur_adresse = new SimpleStringProperty();
        this.administrateur_ville = new SimpleStringProperty();
        this.administrateur_code_postal = new SimpleStringProperty();
        this.administrateur_derniere_connexion = new SimpleStringProperty();
        this.administrateur_date_ajout = new SimpleStringProperty();
    }

    public void setAdministrateur_id(IntegerProperty administrateur_id){this.administrateur_id = administrateur_id;}
    public void setAdministrateur_nom(StringProperty administrateur_nom){this.administrateur_nom = administrateur_nom;}
    public void setAdministrateur_prenom(StringProperty administrateur_prenom){this.administrateur_prenom = administrateur_prenom;}
    public void setAdministrateur_identifiant(StringProperty administrateur_identifiant){this.administrateur_identifiant = administrateur_identifiant;}
    public void setAdministrateur_mot_de_passe(StringProperty administrateur_mot_de_passe){this.administrateur_mot_de_passe = administrateur_mot_de_passe;}
    public void setAdministrateur_email(StringProperty administrateur_email){this.administrateur_email = administrateur_email;}
    public void setAdministrateur_telephone(StringProperty administrateur_telephone){this.administrateur_telephone = administrateur_telephone;}
    public void setAdministrateur_adresse(StringProperty administrateur_adresse){this.administrateur_adresse = administrateur_adresse;}
    public void setAdministrateur_ville(StringProperty administrateur_ville){this.administrateur_ville = administrateur_ville;}
    public void setAdministrateur_code_postal(StringProperty administrateur_code_postal){this.administrateur_code_postal = administrateur_code_postal;}
    public void setAdministrateur_derniere_connexion(StringProperty administrateur_derniere_connexion){this.administrateur_derniere_connexion = administrateur_derniere_connexion;}
    public void setAdministrateur_date_ajout(StringProperty administrateur_date_ajout){this.administrateur_date_ajout = administrateur_date_ajout;}

    public IntegerProperty getAdministrateur_id_Prop(){return administrateur_id;}
    public StringProperty getAdministrateur_nom_Prop(){return administrateur_nom;}
    public StringProperty getAdministrateur_prenom_Prop(){return administrateur_prenom;}
    public StringProperty getAdministrateur_identifiant_Prop(){return administrateur_identifiant;}
    public StringProperty getAdministrateur_mot_de_passe_Prop(){return administrateur_mot_de_passe;}
    public StringProperty getAdministrateur_email_Prop(){return administrateur_email;}
    public StringProperty getAdministrateur_telephone_Prop(){return administrateur_telephone;}
    public StringProperty getAdministrateur_adresse_Prop(){return administrateur_adresse;}
    public StringProperty getAdministrateur_ville_Prop(){return administrateur_ville;}
    public StringProperty getAdministrateur_code_postal_Prop(){return administrateur_code_postal;}
    public StringProperty getAdministrateur_derniere_connexion_Prop(){return administrateur_derniere_connexion;}
    public StringProperty getAdministrateur_date_ajout_Prop(){return administrateur_date_ajout;}

    public int getAdministrateur_id(){return administrateur_id.get();}
    public void setAdministrateur_id(int administrateur_id){this.administrateur_id.set(administrateur_id);}
    public String getAdministrateur_nom(){return administrateur_nom.get();}
    public void setAdministrateur_nom(String administrateur_nom){this.administrateur_nom.set(administrateur_nom);}
    public String getAdministrateur_prenom(){return administrateur_prenom.get();}
    public void setAdministrateur_prenom(String administrateur_prenom){this.administrateur_prenom.set(administrateur_prenom);}
    public String getAdministrateur_identifiant(){return administrateur_identifiant.get();}
    public void setAdministrateur_identifiant(String administrateur_identifiant){this.administrateur_identifiant.set(administrateur_identifiant);}
    public String getAdministrateur_mot_de_passe(){return administrateur_mot_de_passe.get();}
    public void setAdministrateur_mot_de_passe(String administrateur_mot_de_passe){this.administrateur_mot_de_passe.set(administrateur_mot_de_passe);}
    public String getAdministrateur_email(){return administrateur_email.get();}
    public void setAdministrateur_email(String administrateur_email){this.administrateur_email.set(administrateur_email);}
    public String getAdministrateur_telephone(){return administrateur_telephone.get();}
    public void setAdministrateur_telephone(String administrateur_telephone){this.administrateur_telephone.set(administrateur_telephone);}
    public String getAdministrateur_adresse(){return administrateur_adresse.get();}
    public void setAdministrateur_adresse(String administrateur_adresse){this.administrateur_adresse.set(administrateur_adresse);}
    public String getAdministrateur_ville(){return administrateur_ville.get();}
    public void setAdministrateur_ville(String administrateur_ville){this.administrateur_ville.set(administrateur_ville);}
    public String getAdministrateur_code_postal(){return administrateur_code_postal.get();}
    public void setAdministrateur_code_postal(String administrateur_code_postal){this.administrateur_code_postal.set(administrateur_code_postal);}
    public String getAdministrateur_derniere_connexion(){return administrateur_derniere_connexion.get();}
    public void setAdministrateur_derniere_connexion(String administrateur_derniere_connexion){this.administrateur_derniere_connexion.set(administrateur_derniere_connexion);}
    public String getAdministrateur_date_ajout(){return administrateur_date_ajout.get();}
    public void setAdministrateur_date_ajout(String administrateur_date_ajout){this.administrateur_date_ajout.set(administrateur_date_ajout);}
```

6 DAO

- >  AdministrateurDAO.java
- >  Connect.java
- >  JeuneDAO.java
- >  OffreDAO.java
- >  PartenaireDAO.java

DAO signifie Data Access Object, ces classes permettent d'effectuer des requêtes SQL vers une base de données ces requêtes sont contenu dans différentes méthodes selon leur utilisation.

```
package dao;

import classe.Administrateur;

public class AdministrateurDAO
{
    public static boolean getConnexion(String administrateur_identifiant, String administrateur_mot_de_passe) throws ClassNotFoundException, SQLException
    {
        Connection connexion = Connect.getInstance().getConnection();
        boolean reponse = false;

        String req = "SELECT * FROM administrateur WHERE administrateur_identifiant = ? AND administrateur_mot_de_passe = ?";
        PreparedStatement prepared_statement = connexion.prepareStatement(req);








        prepared_statement.setString(1, administrateur_identifiant);
        prepared_statement.setString(2, administrateur_mot_de_passe);

        ResultSet administrateur = prepared_statement.executeQuery();

        if(administrateur.next())
        {
            reponse = true;
            System.out.println();
        }else
        {
        }

        administrateur.close();
        prepared_statement.close();
        connexion.close();
        return reponse;
    }
}
```

7 CONTROLLER









- >  AdministrateurController.java
- >  ConnexionController.java
- >  Jeune_inscriptionController.java
- >  JeuneController.java
- >  OffreController.java
- >  PartenaireController.java
- >  StatistiquesController.java

Le controller permet comme son nom l'indique de contrôler, ce qu'il contrôle ce sont les vues le controller permet d'attribuer des actions aux composants des vues, boutons, champs de texte ou encore tableaux.

```
@FXML
private void adminForm(ActionEvent actionEvent) throws SQLException, ClassNotFoundException, IOException
{
    if(!administrateur_identifiant.getText().equals("") && !administrateur_mot_de_passe.getText().equals(""))
    {
        boolean empdata = AdministrateurDAO.getConnexion(administrateur_identifiant.getText(), administrateur_mot_de_passe.getText());
        String nom = AdministrateurDAO.getNom(administrateur_identifiant.getText(), administrateur_mot_de_passe.getText());
        administrateur_nom.setText(nom);
        System.out.println(nom);

        if(empdata == true)
        {
            try
            {
                mainPane.getChildren().clear();
                FXMLLoader loader = new FXMLLoader();
                loader.setLocation(Main.class.getClassLoader().getResource("view/Menu.fxml"));
                AnchorPane userFrame = (AnchorPane) loader.load();
                Scene sc = new Scene(userFrame);
                Stage stage = (Stage) mainPane.getScene().getWindow();
                stage.setScene(sc);
                System.out.println();
            } catch (IOException e)
            {
                e.printStackTrace();
            }
        }
        else
        {
            Alert a1 = new Alert(Alert.AlertType.ERROR);
            a1.setTitle("Erreur: 2");
            a1.setContentText("Cet utilisateur n'existe pas vérifiez votre mot de passe et votre identifiant.");
            a1.setHeaderText(null);
            a1.showAndWait();
        }
    }
    else
    {
        Alert a1 = new Alert(Alert.AlertType.ERROR);
        a1.setTitle("Erreur: 1");
        a1.setContentText("L'un ou les deux champs sont vide.");
        a1.setHeaderText(null);
        a1.showAndWait();
    }
}
```


8 VUE

-  Administrateur.fxml
-  Connexion.fxml
-  Jeune_inscription.fxml
-  Jeune.fxml
-  Menu.fxml
-  Offre.fxml
-  Partenaire.fxml
-  Statistiques.fxml

Les vue représentent l'interface graphique de l'application.

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.PasswordField?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>

<AnchorPane fx:id="mainPane" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="549.0" prefWidth="882.0" style="-fx-background-color: #223447;" xmlns="http://javafx.com/javafx/8" >
  <children>
    <TextField fx:id="administrateur_identifiant" layoutX="368.0" layoutY="187.0" promptText="Identifiant" style="-fx-background-color: #223447; -fx-border-color: FFFFFF; -fx-text-fill: FFFFFF;" />
    <PasswordField fx:id="administrateur_mot_de_passe" layoutX="368.0" layoutY="249.0" promptText="Mot de passe" style="-fx-background-color: #223447; -fx-border-color: FFFFFF; -fx-text-fill: FFFFFF;" />
    <ImageView fitHeight="36.0" fitWidth="42.0" layoutX="317.0" layoutY="183.0" pickOnBounds="true" preserveRatio="true">
      <image>
        <Image url="@images/contact.png" />
      </image>
    </ImageView>
    <ImageView fitHeight="42.0" fitWidth="48.0" layoutX="314.0" layoutY="241.0" pickOnBounds="true" preserveRatio="true">
      <image>
        <Image url="@images/Lock.png" />
      </image>
    </ImageView>
    <Button fx:id="administrateur_connexion" layoutX="398.0" layoutY="330.0" mnemonicParsing="false" onAction="#administrateur_authentification" prefHeight="27.0" prefWidth="87.0" style="-fx-background-color: #0F87FF; -fx-border-color: #000000; -fx-text-fill: FFFFFF;" />
    <Label alignment="CENTER" layoutX="366.0" layoutY="58.0" prefHeight="71.0" prefWidth="151.0" text="PPE 1" textFill="#0079ff" />
  </children>
</AnchorPane>
```

9 GESTION DES ERREURS

La gestion des erreurs s'effectue dans les controller dans les else des ifs. Si la condition n'est pas respectée le else fera apparaitre une nouvelle fenêtre contenant un message inscrit dans le else.

```
private void administrateur_authentification(ActionEvent actionEvent) throws SQLException, ClassNotFoundException, IOException
{
    if(!administrateur_identifiant.getText().equals("") && !administrateur_mot_de_passe.getText().equals(""))
    {
        boolean empdata = AdministrateurDAO.getConnexion(administrateur_identifiant.getText(), administrateur_mot_de_passe.getText());
        String administrateur_nom = AdministrateurDAO.getNom(administrateur_identifiant.getText(), administrateur_mot_de_passe.getText());

        if(empdata == true)
        {
            try
            {
                mainPane.getChildren().clear();
                FXMLLoader loader = new FXMLLoader();
                loader.setLocation(Main.class.getClassLoader().getResource("view/Menu.fxml"));
                AnchorPane userFrame = (AnchorPane) loader.load();
                Scene sc = new Scene(userFrame);
                Stage stage = (Stage) mainPane.getScene().getWindow();
                stage.setScene(sc);
                System.out.println();

                AdministrateurController administrateur_controller = loader.<AdministrateurController>getController();
                administrateur_controller.setAdministrateur_nom(administrateur_nom);

                JeuneController jeune_controller = loader.<JeuneController>getController();
                jeune_controller.setAdministrateur_nom(administrateur_nom);
            } catch (IOException e)
            {
                e.printStackTrace();
            }
        }
        else
        {
            Alert a1 = new Alert(Alert.AlertType.ERROR);
            a1.setTitle("Erreur: 2");
            a1.setContentText("Cet utilisateur n'existe pas vérifiez votre mot de passe et votre identifiant.");
            a1.setHeaderText(null);
            a1.showAndWait();
        }
    }
    else
    {
        Alert a1 = new Alert(Alert.AlertType.ERROR);
        a1.setTitle("Erreur: 1");
        a1.setContentText("L'un ou les deux champs sont vide.");
        a1.setHeaderText(null);
        a1.showAndWait();
    }
}
```

