



ROYAUME DU MAROC
UNIVERSITÉ ABDELMALEK ESSAADI
FACULTÉ DES SCIENCES ET TECHNIQUE
TANGER



POO en JAVA

Mini-Projet : Interface Graphique



GROUPE : 2

Réalisée par :

wiam charkaoui

Encadrée par :

El Mokhtar EN-NAIMI

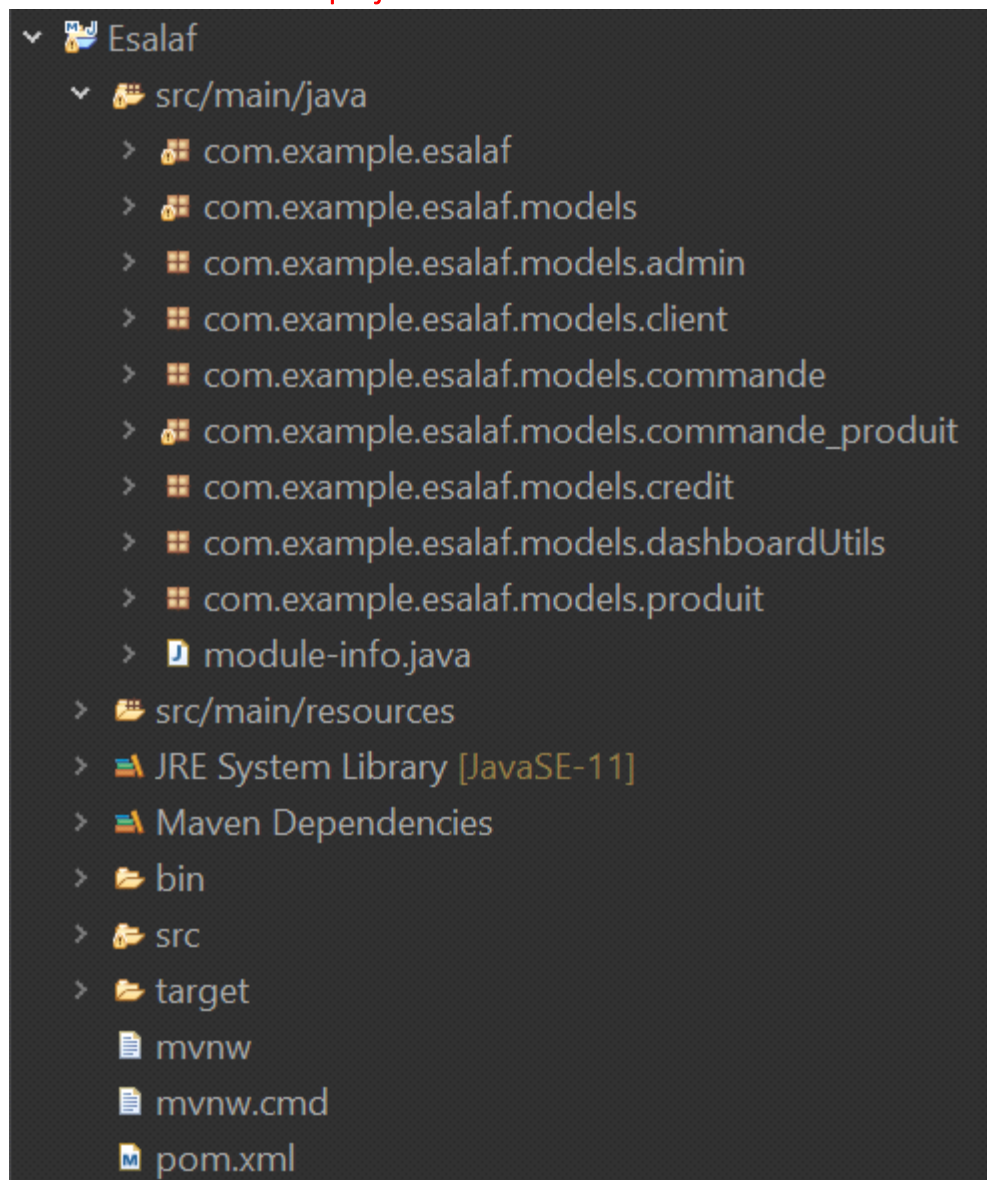
LOTFI Elaachak

Introduction :

Interface graphique :

- Une interface graphique est formée d'une ou plusieurs fenêtres qui contiennent divers composants graphiques (widgets) tels que : boutons, listes déroulantes, menus, champ texte, etc.
- Les interfaces graphiques sont souvent appelés GUI d'après l'anglais « Graphical User Interface »

Structure du projet :



Installation du projet :

Configuration du projet :

- 1 - Extraire le fichier .rar
- 2 - ouvrir IDE (Eclipse ou IntelliJ IDEA)
- 3 - click sur Open projet -> choisir le dossier du projet (le dossier qui a été extrait depuis le fichier rar)
- 4 - pour lancer le projet : ouvrir et lancer le fichier LoginStage.java.

Installation de la base de données :

- 1 - ouvrir votre éditeur de base de données
- 2 - créer une nouvelle base de données et nommer la "esalaf " ou n'importe quel nom tu préfères.
- 3 -En cas d'un nom différent a « esalaf » :
Paramétrer le fichier Base DAO dans le package ./esalaf/models pour vous connecter à la base de donnée
- 4 - toujours dans votre éditeur de base de données lancer ce script SQL :

Application :

Interface Client

[illegible]

Ajouter client :

Code :

```
@FXML
void btnAjouterAction(ActionEvent event) throws SQLException {
    errorLabel.setText("");
    if (validate(nomClientText.getText(), type: "nom") && validate(telephoneClientText.getText(), type: "telephone")) {
        ClientDAO clientDAO = new ClientDAO();
        Client client = new Client(nomClientText.getText(), telephoneClientText.getText());
        try {
            clientDAO.save(client);
            updateClientTable();
            btnClrAction(event);
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Interface :

Code :

```
@FXML
void btnModifierAction(ActionEvent event) throws SQLException {
    errorLabel.setText("");
    if (validate(nomClientText.getText(), type: "nom") &&
        validate(telephoneClientText.getText(), type: "telephone") &&
        validate(text: "-", type: "tableselected")) {
        ClientDAO clientDAO = new ClientDAO();
        Client client = new Client(id, nomClientText.getText(), telephoneClientText.getText());
        try {
            clientDAO.update(client);
            updateClientTable();
            btnClrAction(event);
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```










Interface :

[illegible]

Résultat :

ID Client	Nom Complet	Telephone	
1234	client one	444442222	
1235	client two	11223344	
1236	client three	111113333322	

Base de données :

Extra options				id_client	nomcomplet_client	telephone_client
<input type="checkbox"/>	 Edit	 Copy	 Delete	1234	client one	444442222
<input type="checkbox"/>	 Edit	 Copy	 Delete	1235	client two	11223344
<input type="checkbox"/>	 Edit	 Copy	 Delete	1236	client three	1111113333322

Supprimer client :

Code :

```
@FXML
void btnSupprimerAction(ActionEvent event) throws SQLException {
    errorLabel.setText("");
    if (validate(text: "-", type: "tableselected")) {
        ClientDAO clientDAO = new ClientDAO();
        Client client = new Client(id, nomcomplet_client: null, telephone_client: null);
        try {
            clientDAO.delete(client);
            updateClientTable();
            btnClrAction(event);
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Interface :

[illegible]

Résultat :

Client

Nom Complet :

Telephone :

ID Client	Nom Complet	Telephone	
1234	client one	444442222	
1235	client two	11223344	

Base de données :

Extra options

↔T↔

▼

id_client

nomcomplet_client

telephone_client

☐

 Edit

 Copy

 Delete

1234

client one

444442222

☐

 Edit

 Copy

 Delete

1235

client two

11223344

Interface Produit :

Produit

Nom produit :

Prix produit :

Ajouter

Modifier

Clear

Supprimer

ID Produit	Nom Produit	Prix Produit	
1	Iphone	2223.0	
2	realme	1245.0	

Ajouter Produit :

Code :

@FXML










Interface :

[illegible]

Résultat :

ID Produit	Nom Produit	Prix Produit	
1	Iphone	2223.0	
2	realme	1245.0	
3	OPPO	999.0	

Base de données :

<input type="checkbox"/>	 Edit	 Copy	 Delete	1 Iphone	2223
<input type="checkbox"/>	 Edit	 Copy	 Delete	2 realme	1245
<input type="checkbox"/>	 Edit	 Copy	 Delete	3 OPPO	999

Modifier Produit :

Code :

```
@FXML
void btnModifierProduitAction(ActionEvent event) throws SQLException {
    errorLabel.setText("");
    ProduitDAO produitDAO = new ProduitDAO();
    if (validate(nomProduitText.getText(), type: "nom") &&
        validate(prixProduitText.getText(), type: "prix") &&
        validate(text: "-", type: "tableselected")) {
        Produit produit = new Produit(id, nomProduitText.getText(), Double.parseDouble(prixProduitText.getText()));
        produitDAO.update(produit);
        updateTable();
        btnClrProduitAction(event);
    }
}
```










Interface :

[illegible]

Résultat :

	ID Produit	Nom Produit	Prix Produit	
	1	Iphone	2223.0	
	2	realme	445.0	
	3	OPPO	999.0	

Base de données :

					id_produit	nom_produit	prix_produit
<input type="checkbox"/>		Edit		Copy		Delete	1 Iphone 2223
<input type="checkbox"/>		Edit		Copy		Delete	2 realme 445
<input type="checkbox"/>		Edit		Copy		Delete	3 OPPO 999

Supprimer Produit :

Code :

```
@FXML
void btnSupprimerProduitAction(ActionEvent event) throws SQLException {
    ProduitDAO produitDAO = new ProduitDAO();
    if (validate( text: "-", type: "tableselected")) {
        produitDAO.delete(new Produit(id, nom_produit: null, prix_produit: 0));
        updateTable();
        btnClrProduitAction(event);
    }
}
```

Interface :

[illegible]

Client :

Date Commande :

Produits :

1

Ajouter produit

Ajouter

Modifier

Clear

Sert à présenter la relation entre le client et un ou plusieurs produits dans le process d'achats ou de réservation, l'interface offre la possibilité d'ajouter une nouvelle commande pour un client, soit par un seul produit, ou plusieurs, d'un seul type ou plusieurs

Code :

Interface :

[illegible]

Client :

Date Commande :

4/29/2023

Produits :

ID Commande	Client		Prix Total	
1	client one	2023-03-08	138.0	
2	client one	2023-03-09	414.0	
3	client two	2023-04-05	1271.0	
4	client two	2023-04-29	10781.0	

Code :

```
@FXML
void btnModifierCommandeAction(ActionEvent event) throws SQLException {
    if (id != null) {
        if (validate( type: "commande")) {
            Commande commande = new Commande(id, clientsMap.get(clientBox.getValue()), java.sql.Date.valueOf(dateCommande.getValue()), coutTotal);
            CommandeDAO commandeDAO = new CommandeDAO();
            try {
                commandeDAO.update(commande);
                Commande_ProduitDAO commande_produitDAO = new Commande_ProduitDAO();
                Commande_Produit commande_produit;
                for (int i = 0; i < produitsSelected.size(); i++) {
                    commande_produit = new Commande_Produit(commandeDAO.getLast(), produitsSelected.get(i), produitsSelectedQuantite.get(i));
                    commande_produitDAO.save(commande_produit);
                }
                btnClrCommandeAction( event: null);
                updateTable();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
}
```

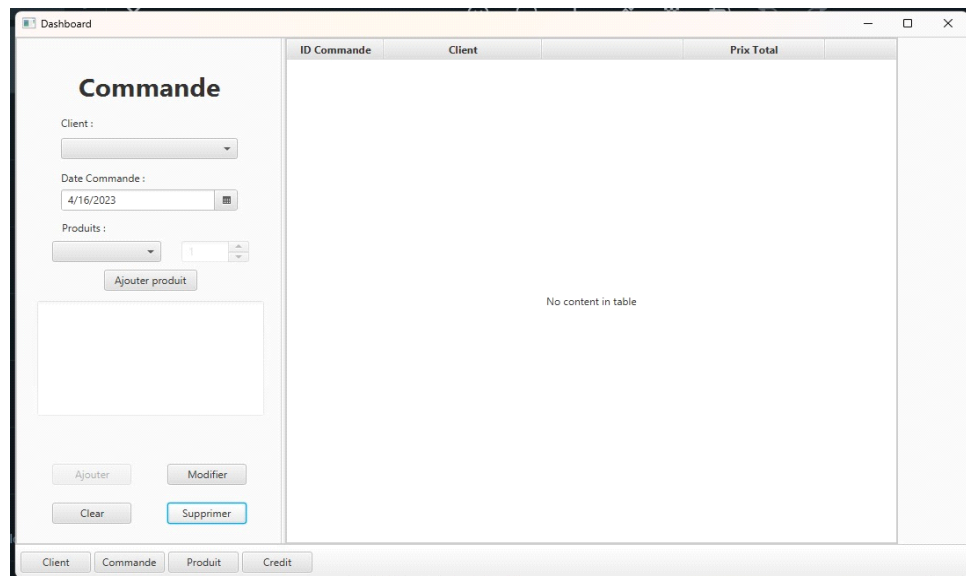
[illegible]

Supprimer Commande :

Code :

```
@FXML
void btnSupprimerCommandeAction(ActionEvent event) throws SQLException {
    if (id != null) {
        CommandeDAO commandeDAO = new CommandeDAO();
        try {
            commandeDAO.delete(new Commande(id, id_client: null, date_commande: null, prix: 0));
            btnClrCommandeAction(event: null);
            updateTable();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Interface :



Interface Crédit :

[illegible]

Client :

Commande

Date Credit :

Page 10 of 10

Ajouter

Modifie

Clear

Sert à présenter la relation entre le client et une commande

Code :

Interface :

[illegible]

Modifier Crédit :

Code :

```
@FXML
void btnModifierCreditAction(ActionEvent event) throws SQLException {
    if (validate( type: "client") && validate( type: "commande") && validate( type: "date")) {
        Credit credit = new Credit(id,clientsMap.get(clientBox.getSelectionModel().getSelectedItem()),
            commandesMap.get(commandeBox.getSelectionModel().getSelectedItem()),
            java.sql.Date.valueOf(dateFieldCredit.getValue()));
        CreditDAO creditDAO = new CreditDAO();
        creditDAO.update(credit);
        btnClrCreditAction(event);
        updateTable();
    }
}
```

Interface :

[illegible]

Client :

client two

Commande :

2023-04-29 : 10781.0

Date Credit :

4/7/2023

Iphone : 2223.0x 4
 realme : 445.0x 2
 OPPO : 999.0x 1
 Total : 10781.0

Veillez choisir une co...

Ajouter

Modifier

Clear

Supprimer

Supprimer Crédit :

Code :

@FXML

Interface :

Credit

Client :

Commande :

Date Credit :

Ajouter

Modifier

Clear