

# RAPPORT DE MINI PROJET JAVA

Gestion de facturation



**Encadré par :**

Mr. El Mokhtar EN-NAIMI

Mr. Lotfi ELAACHAK

**Réalisé par :** *Israe SAIDI*

**Groupe :** 2-1

## Objectif :

Le principe de ce mini-projet est la création d'une application qui sert à faciliter les différentes tâches effectuées par les vendeurs (la gestion de clients, la gestion de produits, l'enregistrement des commandes et le remplissage de Factures).

## Introduction :

Afin de réaliser ce mini-projet on a utilisé JAVA **Swing**.

Java Swing fait partie de Java Foundation Classes (JFC) qui est utilisé pour créer des applications basées sur des fenêtres. Il est construit sur le dessus de l'API AWT (Abstract Windowing Toolkit) et entièrement écrit en java.

Le package javax.swing fournit des classes pour l'API java swing telles que JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.

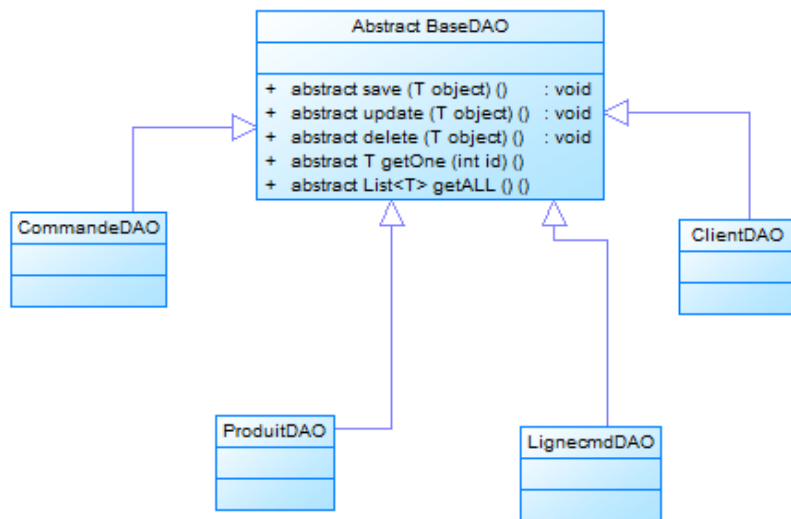
**NB : La fonction main existe dans « Home.java »**

```
public static void main(String[] args) {  
    EventQueue.invokeLater(new Runnable() {  
        public void run() {  
            try {  
                Home frame = new Home();  
                frame.setVisible(true);  
            } catch (Exception e) {  
                e.printStackTrace();  
            }  
        }  
    });  
}
```

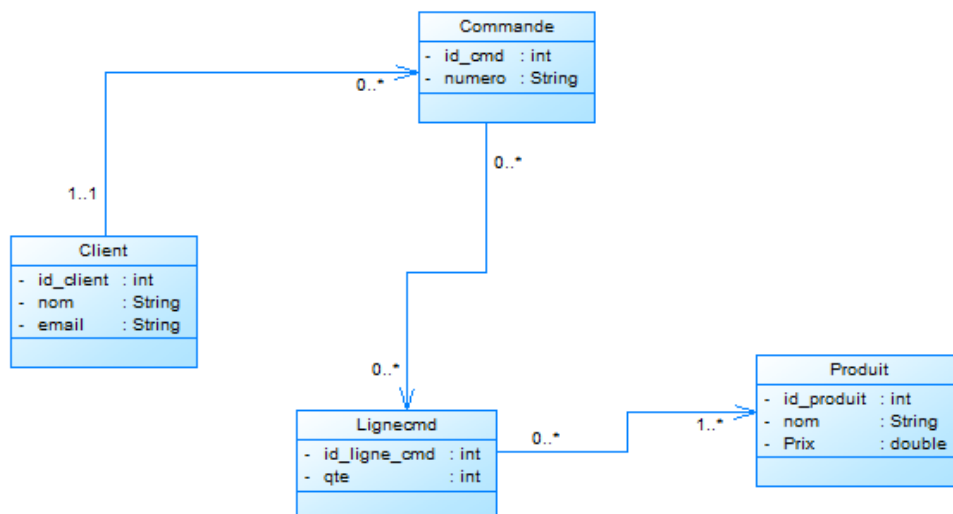
## Partie 1 : Conception

Dans le dossier de ce mini-projet il existe 3 packages :

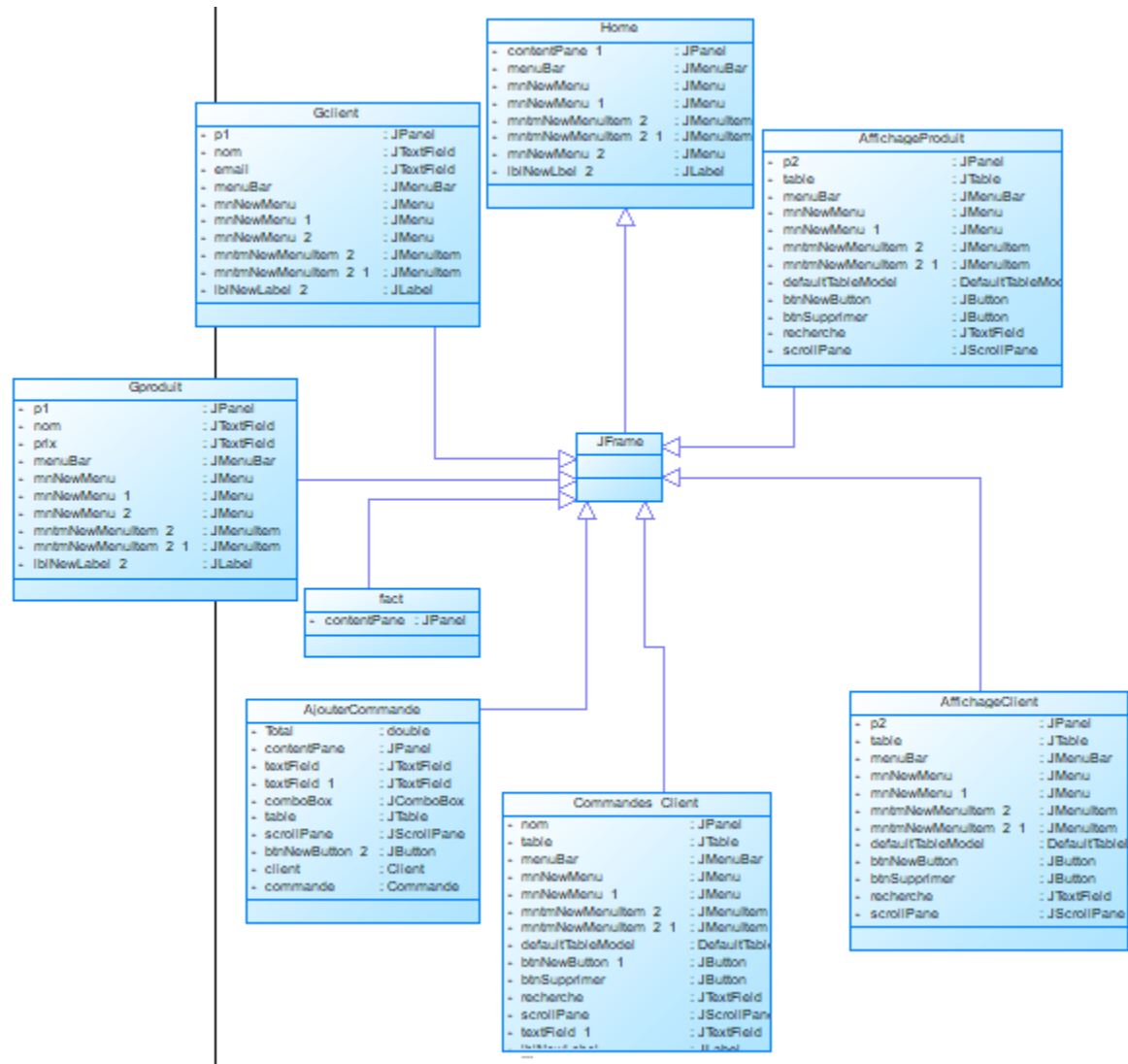
**package** ma.fstt.dao;



**package** ma.fstt.model




package ma.fstt.view




Créer une base de données sous le nom **gfacture**, après la migration des tables on aura cette architecture :

Table	Action
client	★ Parcourir Structure Rechercher Insérer Vider Supprimer
cmd	★ Parcourir Structure Rechercher Insérer Vider Supprimer
lign_cmd	★ Parcourir Structure Rechercher Insérer Vider Supprimer
produit	★ Parcourir Structure Rechercher Insérer Vider Supprimer
4 tables	Somme



### La table client :

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra
1	id_client 	int(11)			Non	Aucun(e)		AUTO_INCREMENT
2	nom	varchar(255)	utf8mb4_general_ci		Oui	NULL		
3	email	varchar(255)	utf8mb4_general_ci		Oui	NULL		


### La table produit :

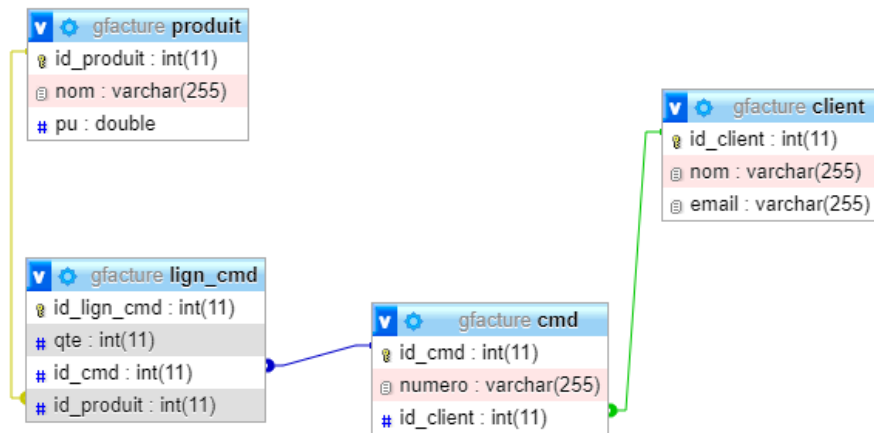
#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra
1	id_produit 	int(11)			Non	Aucun(e)		AUTO_INCREMENT
2	nom	varchar(255)	utf8mb4_general_ci		Oui	NULL		
3	pu	double			Oui	NULL		

### La table cmd :

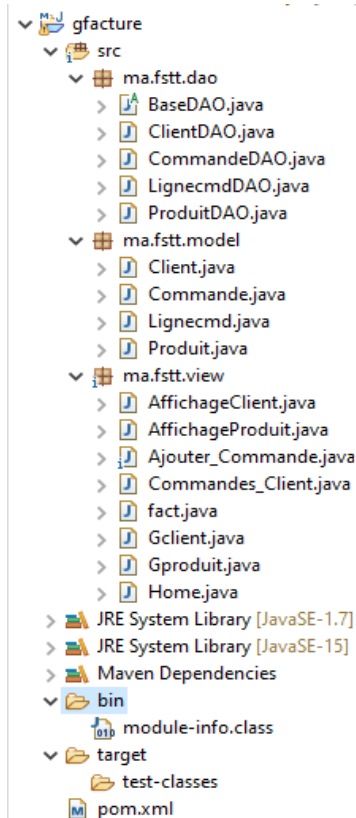
#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra
1	id_cmd 	int(11)			Non	Aucun(e)		AUTO_INCREMENT
2	numero	varchar(255)	utf8mb4_general_ci		Non	Aucun(e)		
3	id_client 	int(11)			Non	Aucun(e)		

### La table lign\_cmd :

1	id_lign_cmd 	int(11)			Non	Aucun(e)		AUTO_INCREMENT
2	qte	int(11)			Oui	NULL		
3	id_cmd 	int(11)			Non	Aucun(e)		
4	id_produit 	int(11)			Non	Aucun(e)		



## L'architecture du projet :



Il faut ajouter les dépendances dans le fichier « **porm.xml** »

```
<dependencies>

  <!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.23</version>
  </dependency>

</dependencies>
```

Et afin d'établir la connexion avec la base de données « **MySql** »

```
public abstract class BaseDAO <T>    {

    protected Connection connection ;

    protected Statement statement ;

    protected PreparedStatement preparedStatement ;

    protected ResultSet resultset ;

    private String url = "jdbc:mysql://localhost:3306/gfacture";
    private String user = "root";

    private String password = "";

    public BaseDAO() throws SQLException {

        this.connection = DriverManager.getConnection(url, user, password);
    }
}
```

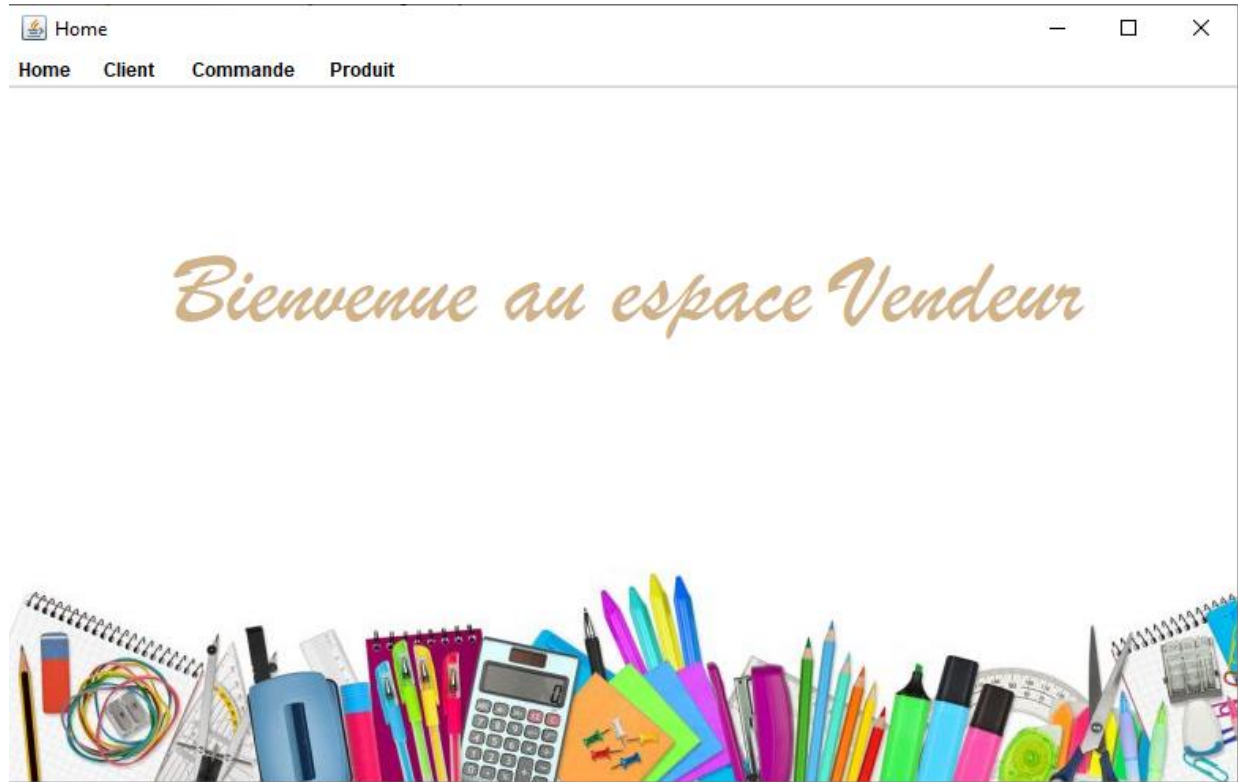
Voici la classe abstraite « **BaseDAO** » qui définit toutes les méthodes de gestion de facturation (l'implémentation se fait dans les **classes** filles)

```
public abstract class BaseDAO <T> {  
  
    protected Connection connection ;  
  
    protected Statement statement ;  
  
    protected PreparedStatement prepareStatement ;  
  
    protected ResultSet resultSet ;  
  
    private String url = "jdbc:mysql://localhost:3306/gfacture";  
    private String user = "root";  
  
    private String password = "";  
  
    public BaseDAO() throws SQLException {  
  
        this.connection = DriverManager.getConnection(url, user, password);  
    }  
  
    public abstract void save (T object) throws SQLException ;  
    public abstract void update (T object) throws SQLException ;  
    public abstract void delete (T object) throws SQLException ;  
    public abstract T getOne (int id) throws SQLException ;  
    public abstract List<T> getALL () throws SQLException ;  
}
```



## Partie 2 : Réalisation

La fenêtre Home :

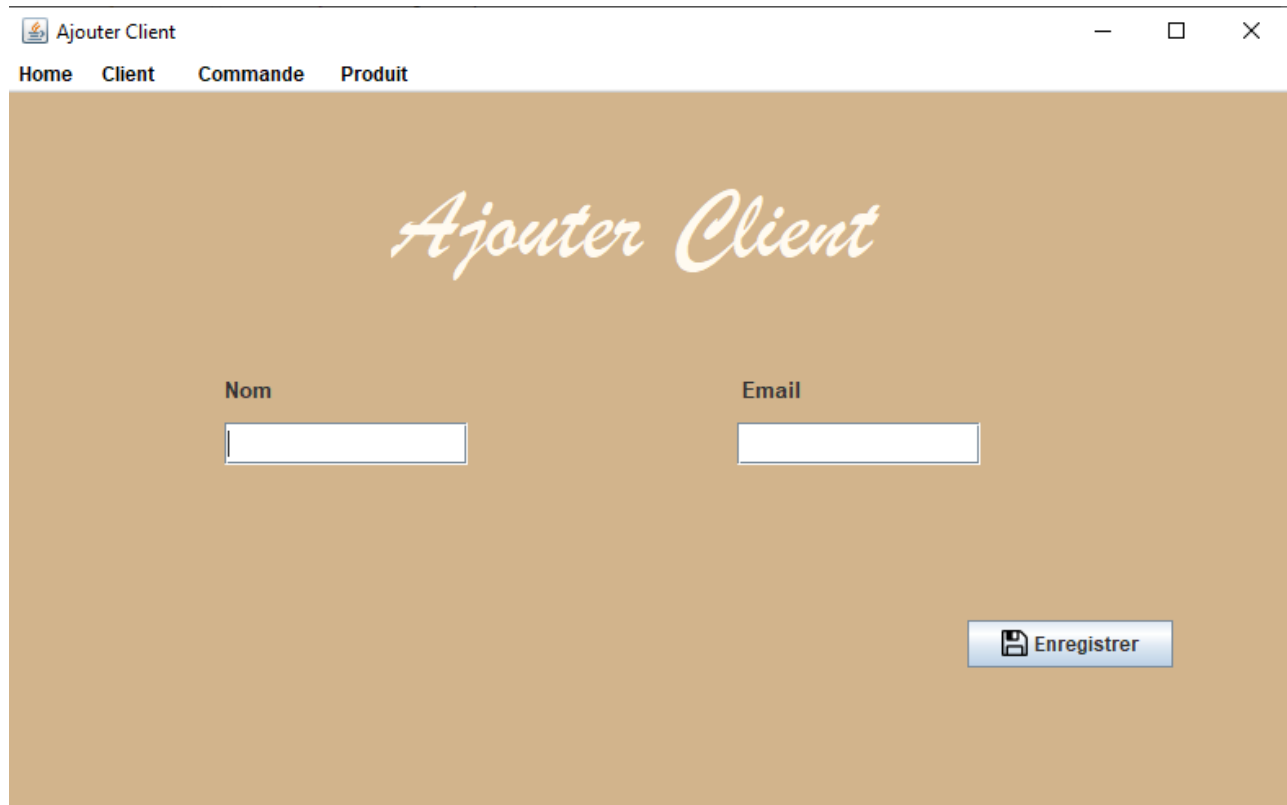


Le Menu Barre :



### La fenêtre Ajouter Client :

Afin d'ajouter des nouveaux clients il faut obligatoirement remplir les deux champs (**Nom** et **Email**) et effectuer un clic sur le bouton **Enregistrer**.



Ajouter Client

Home Client Commande Produit


Nom

Email

Enregistrer

### La fenêtre Nos Clients :

Cette fenêtre nous permet d'**afficher** la liste de tous les clients Enregistrer et il nous offre aussi la possibilité de modifier les données de nos clients (effectuer un clic sur le bouton **Modifier**) ou de les Supprimer (effectuer un clic sur le bouton **Supprimer**) et aussi **chercher** un client.

 Nos Clients

[Home](#)
[Client](#)
[Commande](#)
[Produit](#)

# Nos Clients


id_client	nom	email
1	Ist	Ist@uae.ac.ma
2	Istisrae	Ist2@uae.ac.ma
8	israe	israesd@gmail.com
9	laila	laila@gmail.com
10	younes	younes@gmail.com

[Home](#)
[Client](#)
[Commande](#)
[Produit](#)

Ajouter un nouveau produit  
 afficher tous les produits

### La fenêtre Ajouter Produit :

C'est le même principe que la fenêtre « Ajouter Client », pour qu'on puisse ajouter un produit dans notre liste il faut obligatoirement remplir les deux champs (**Nom** et **Prix**) et effectuer un clic sur le bouton **Enregistrer**.

Ajouter produit

—

□

×

Home

Client

Commande

Produit

Ajouter Produit


Nom

prix

Enregistrer

### La fenêtre Nos Produits :

Le même fonctionnement que la fenêtre **Nos Clients**.

Nos Produits

—

□

×

Home

Client

Commande

Produit

Nos Produits

Recherche

id_produit	nom	prix
1	Stylo	3.0
2	Gomme	10.0
3	correcteur	10.0
4	Calculatrice	120.0
5	Cahier	10.0
6	crayon	2.0
7	feutre	16.0

Modifier

Supprimer

### La fenêtre Voir Commandes :


Cette fenêtre nous permet d'afficher les commandes d'un tel client (on tape **id\_client** dans la zone de texte et on effectue un clic sur **Voir Détails**), Ajouter des nouvelles commandes à l'aide du bouton **Ajouter Commande**, Supprimer une commande et aussi afficher les détails des commandes enregistrés en cliquant sur le bouton **Voir facture**.


Voir Commandes

Home Client **Commande** Produit




# Commandes

id\_client

9  Voir Détails

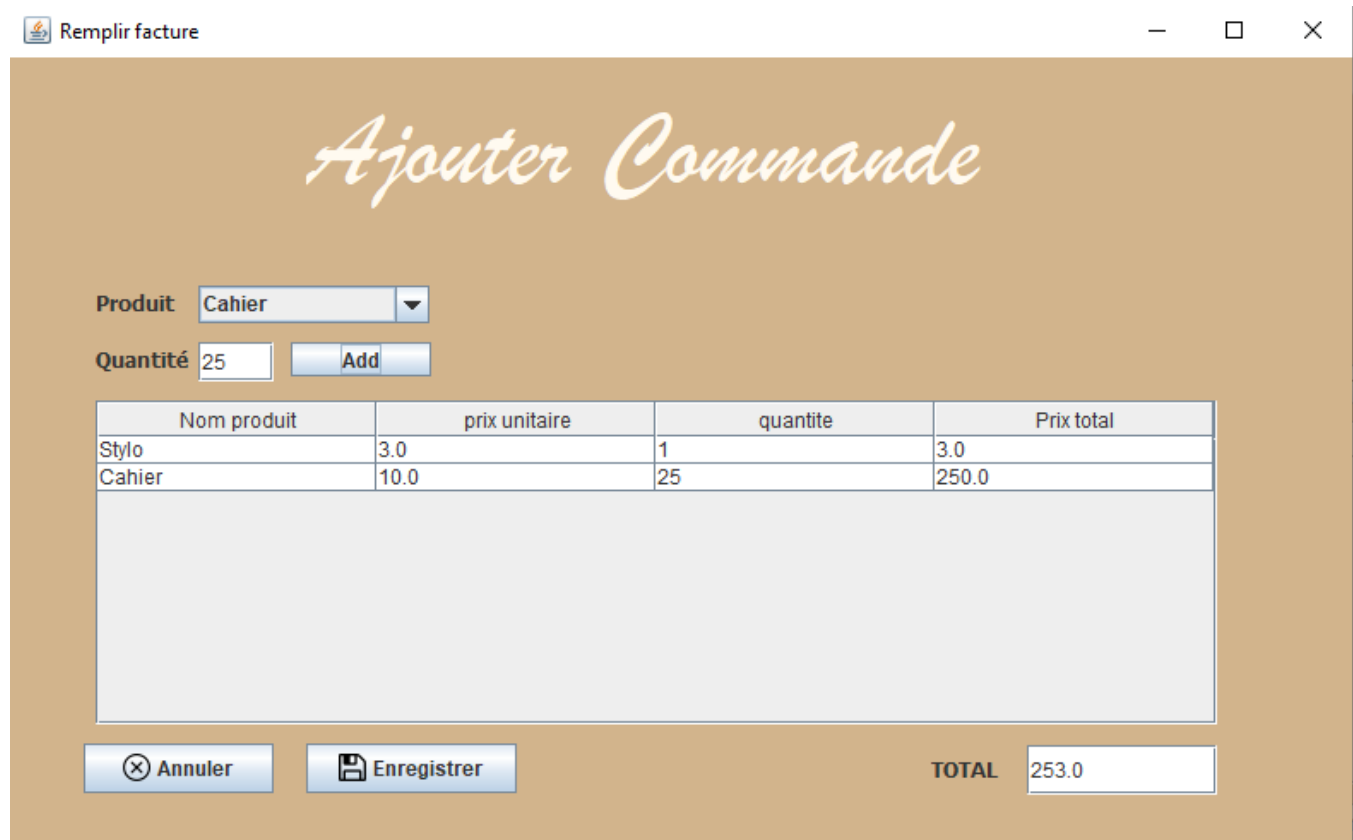
 Recherche

id	numero
4	5500
5	556

 Ajouter Comman...  Voir facture  Supprimer

### La fenêtre Remplir facture :

Cette fenêtre nous donne la possibilité d'ajouter des lignes de commandes (produit et quantité) et calculer le montant total de cette commande.




The screenshot shows a window titled "Remplir facture" with a standard Windows title bar (minimize, maximize, close buttons). The main area has a tan background and features the text "Ajouter Commande" in a large, elegant script font. Below this, there are input fields for "Produit" (a dropdown menu showing "Cahier") and "Quantité" (a text box with "25"). An "Add" button is positioned next to the quantity field. A table with four columns is displayed: "Nom produit", "prix unitaire", "quantite", and "Prix total". The table contains two rows: "Stylo" with a unit price of 3.0 and quantity of 1, and "Cahier" with a unit price of 10.0 and quantity of 25. Below the table is a large, empty rectangular area. At the bottom of the window, there are two buttons: "Annuler" (with a close icon) and "Enregistrer" (with a save icon). To the right of these buttons, the word "TOTAL" is displayed next to a text box containing the value "253.0".

Nom produit	prix unitaire	quantite	Prix total
Stylo	3.0	1	3.0
Cahier	10.0	25	250.0

Et après le remplissage de notre commande on peut soit Annuler la commande (Bouton **Annuler**) ou l'enregistrer et afficher Facture à l'aide du bouton **Enregistrer**.

## La fenêtre Facture :

 Facture

**Votre Facture**

Sun May 02 04:38:40 WET 2021

Numero de Commande : 5500

Nom de Client : laila

Produit	Quantité	Prix unitaire
Stylo	1	3.0
crayon	10	2.0

Montant à payer : 23.0 MAD

Merci pour Votre Confiance :)