

# RAPPORT D'APPLICATION JAVA

Projet réalisé par : Oumaima Ettassouli

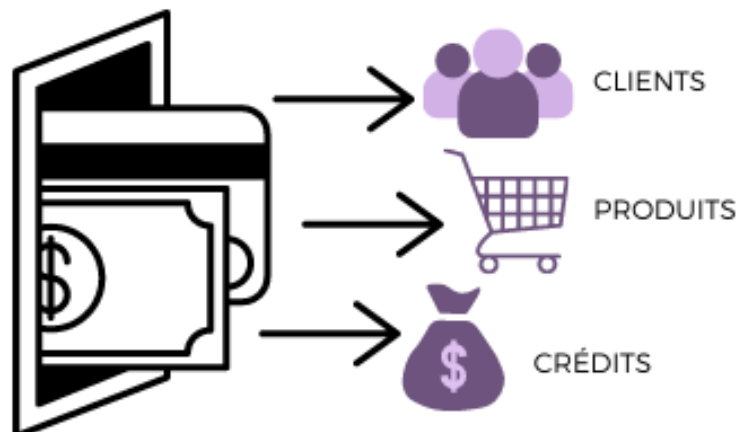
Encadré par : Prof. Lelaachak Lotfi Et Prof. El Mokhtar En-naimi



OUTILS



GÉRER VOS CRÉDITS EN TOUTE SIMPLICITÉ



CONTRÔLEZ VOS CRÉDITS,  
OPTIMISEZ VOTRE GESTION,  
GAGNEZ DU TEMPS !

# Présentation du projet

## Let's take a closer look at the application!

Ce projet est sous forme d'une application desktop Java destinée à moderniser la manière dont les crédits des clients sont enregistrés. Contrairement à la méthode traditionnelle qui utilise un carnet, cette application offre une expérience plus fluide et plus agréable pour les utilisateurs.

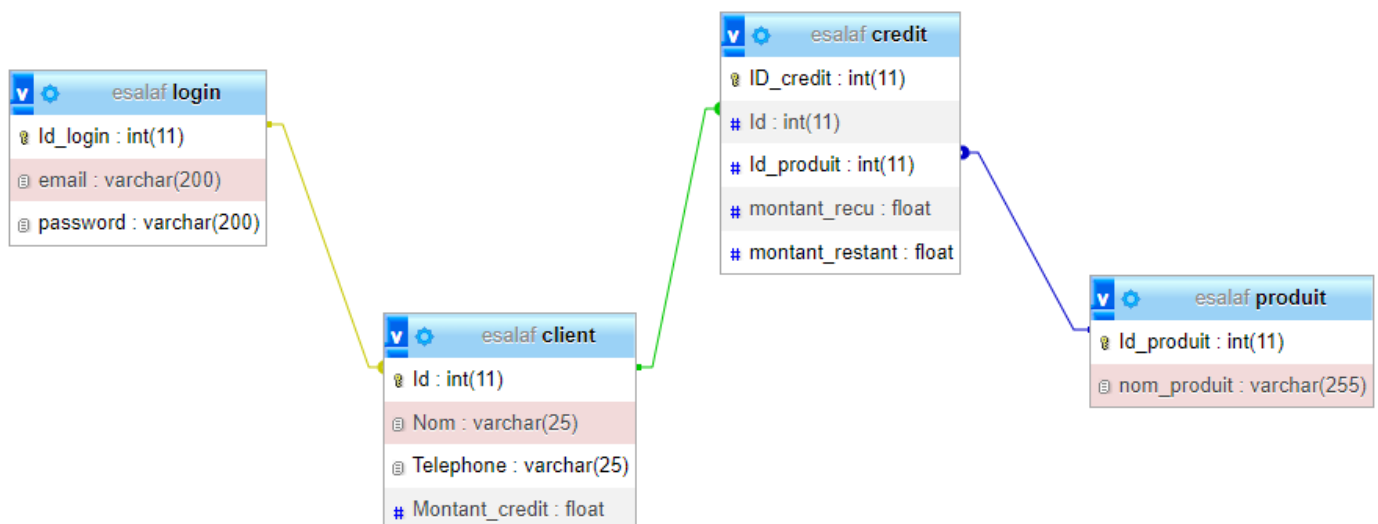
Ils peuvent enregistrer et gérer les crédits des clients : ajouter de nouveaux clients, saisir et modifier les différentes informations relatives à leurs crédits (Montant de crédit total, Montant reçu, Montant restant à payer...) ainsi que les produits achetés.

### Architecture technique:

Concernant l'interface graphique de l'application, j'ai utilisé JavaFx qui permet de créer des interfaces utilisateur riches et interactives offrant ainsi, des fonctionnalités avancées telles que l'animation, les effets visuels...

Pour stocker les différentes informations relatives aux clients et aux transactions, j'ai utilisé **Java JDBC** pour communiquer avec la base de données « esalaf » MySQL.

Schéma de la base de données :



Il y avait également l'utilisation du pattern **Model-View-Controller (MVC)** pour organiser le code de manière logique et cohérente.

---

# SOMMAIRE

## I. Présentation de l'application

## II. Conception

1. Modèles – Controllers- Views
2. Ressources
3. Expérience personnelle

## III. Etude du projet

1. Authentification
2. Tableau de bord
3. Gestion des clients
4. Déclaration des achats
5. Gestion des crédits

## IV. Annexes

### Fonctions CRUD

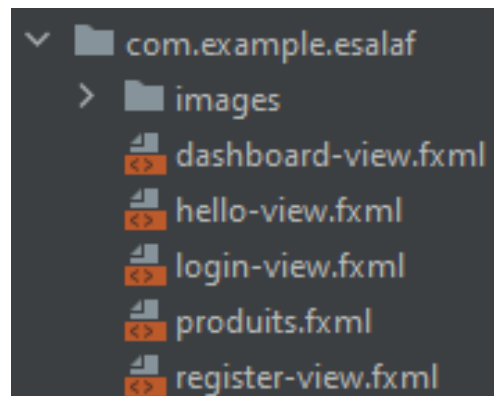
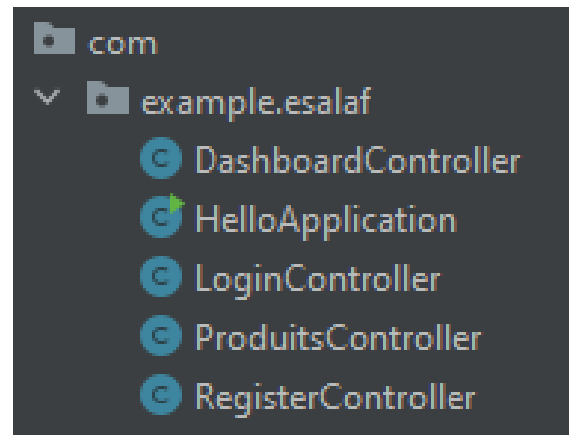
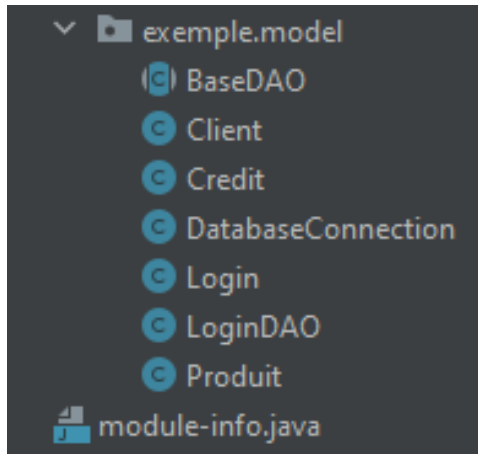
1. Save
2. Update
3. Delete
4. Search

## V. Bibliographie

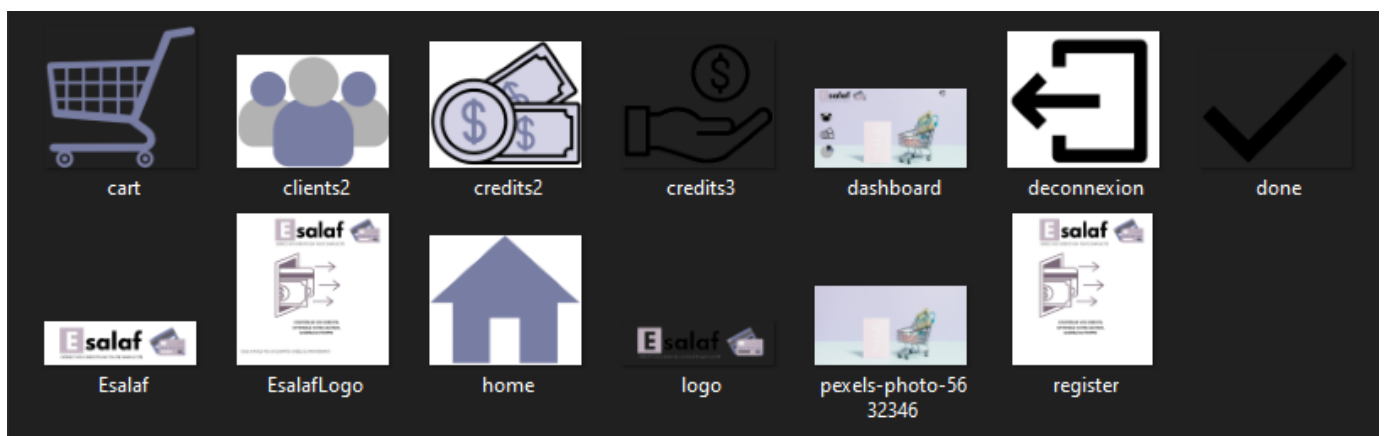
## VI. Remerciements

# Conception

## Les classes – Models, Controllers, Views



## Ressources/ Images



---

## Expérience personnelle

Honnêtement, travailler sur ce projet a été une expérience très gratifiante pour moi, cela m'a aidé à comprendre que je pouvais être mon propre soutien et à compter sur moi-même pour atteindre mes objectifs. Même si cela a été un véritable défi à certains moments.

Parmi les difficultés que j'ai rencontré :

- **L'inexpérience avec la technologie utilisée** : JavaFX utilise une architecture de programmation différente de celle des autres framework d'interface graphique. Il m'a donc fallu du temps pour me familiariser avec la structure et les composants de JavaFX.
- **Conception de l'interface utilisateur** : l'un des plus grands challenges est la compréhension des problématiques que les utilisateurs peuvent rencontrer. Il faut étudier tous les cas possibles, en envisageant toutes les situations dans lesquelles un utilisateur pourrait utiliser l'application.
- **Les erreurs et les bugs qui ne s'arrêtent pas** : c'était trop frustrant de passer des heures devant l'écran essayant de résoudre toutes ces erreurs qui empêchent l'application de fonctionner.

En général, malgré toutes les difficultés qu'on peut rencontrer dans n'importe quel projet, il faut rester positif, persévérant et surtout ne pas abandonner.

# Etude du projet

## 1. Authentification

C'est la première scène qui s'affiche lors du lancement de l'application.

Les utilisateurs ont la possibilité de se connecter s'ils ont déjà un compte ou de créer un nouveau compte.

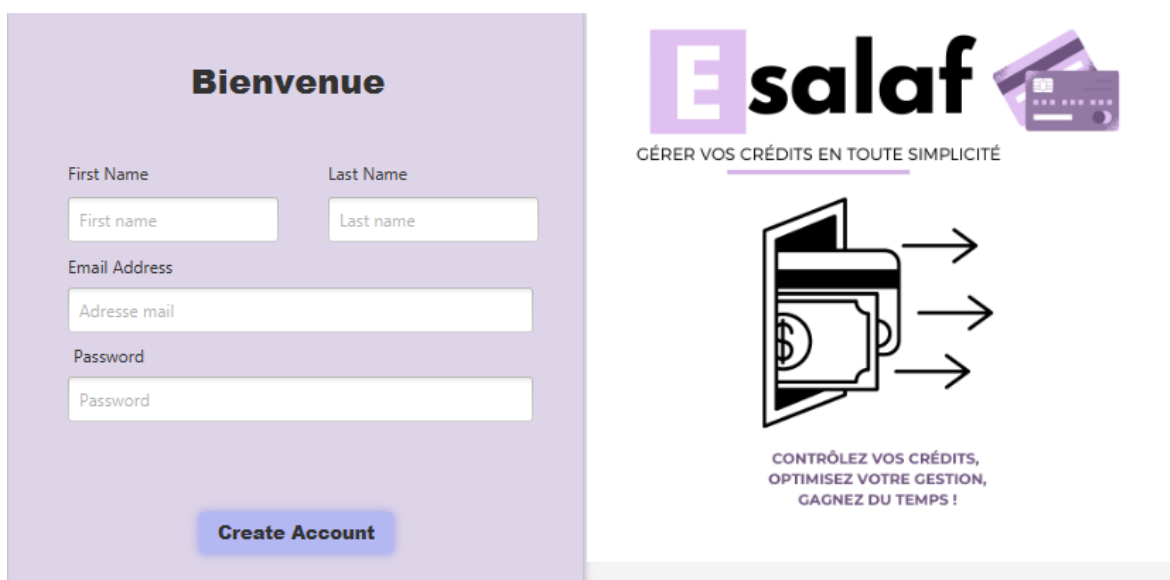
- Pour se connecter, les utilisateurs peuvent utiliser leur adresse e-mail ainsi que leur mot de passe.



The mockup shows a light purple background. On the left, the 'Esalaf' logo is displayed with the tagline 'GÉRER VOS CRÉDITS EN TOUTE SIMPLICITÉ'. Below the logo is an illustration of a wallet with three arrows pointing out, and the text 'CONTRÔLEZ VOS CRÉDITS, OPTIMISEZ VOTRE GESTION, GAGNEZ DU TEMPS !'. At the bottom left, a link says 'VOUS N'AVEZ PAS UN COMPTE? CRÉEZ-LE MAINTENANT!'. To the right, a 'Connexion' section contains two input fields for 'Adresse Email' and 'Mot de passe', and a 'Se Connecter' button. At the bottom right, there is a 'Créer Compte' button.

Sinon,

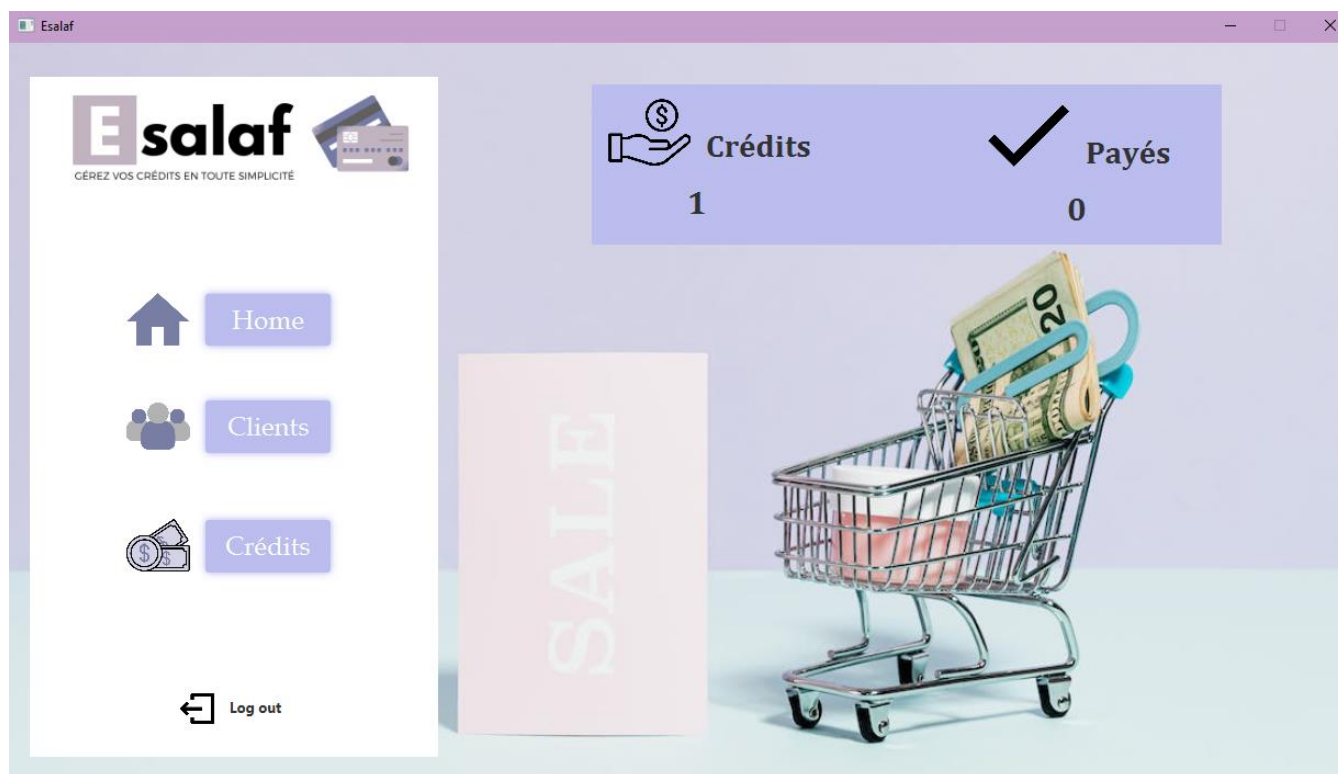
- Créer un nouveau compte en cliquant sur (créer compte) et remplir les informations requises telles que leur nom, prénom, adresse e-mail et mot de passe. Il est important de noter que les utilisateurs doivent fournir des informations valides et de ne pas laisser les champs vides.



The mockup shows a light purple background. On the left, a 'Bienvenue' section contains four input fields: 'First Name' (with placeholder 'First name'), 'Last Name' (with placeholder 'Last name'), 'Email Address' (with placeholder 'Adresse mail'), and 'Password' (with placeholder 'Password'). A 'Create Account' button is at the bottom left. On the right, the 'Esalaf' logo and tagline are shown. Below them is the same wallet illustration and text as in the previous mockup.

## 2. Tableau de bord

Après la confirmation de l'authentification, l'utilisateur se trouve face à un menu où il peut accéder à des fonctionnalités telles que la création de nouveaux crédits, la modification de crédits existants ou la modification des informations des clients.



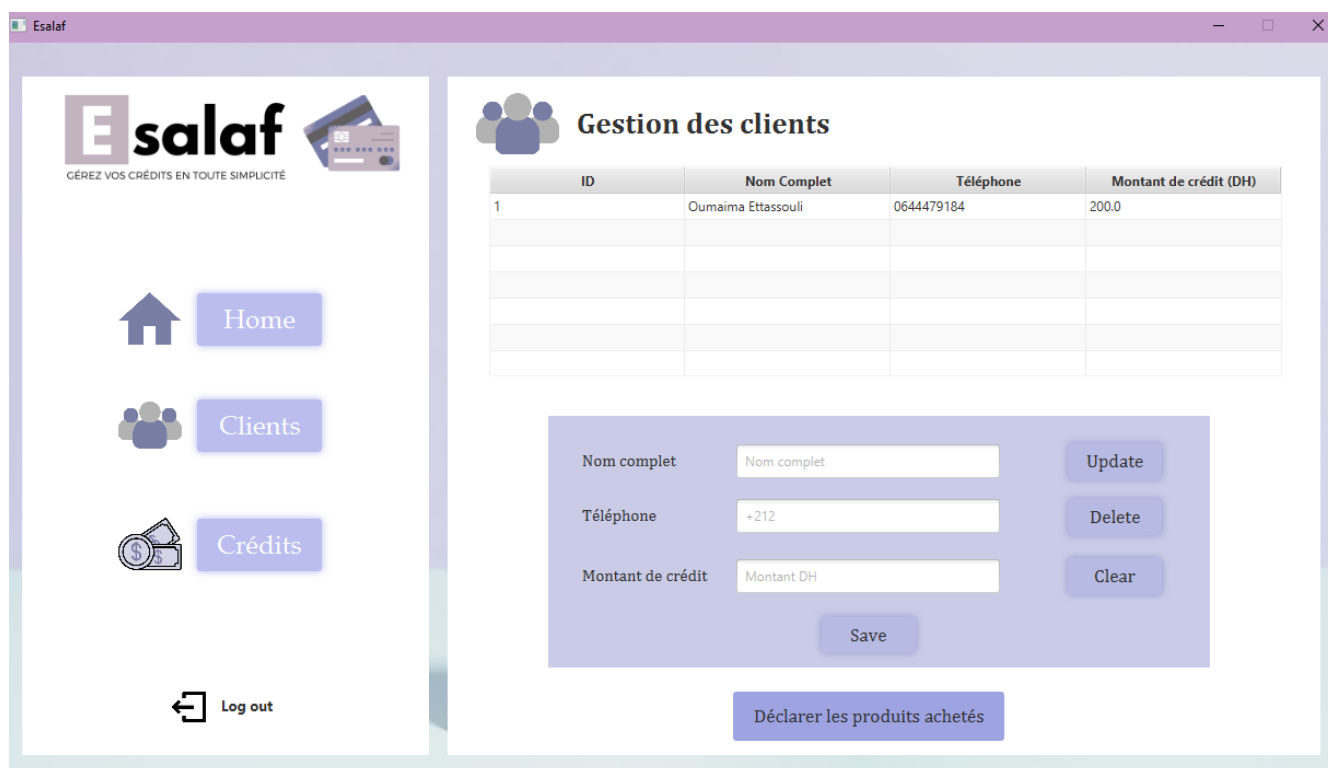
Dans la partie des statistiques du tableau de bord, l'utilisateur peut trouver des informations sur les crédits. Par exemple, il y a le nombre total de crédits disponibles et le nombre de crédits payés par les clients.

L'utilisateur peut aussi se déconnecter en cliquant sur « Log out ».

### 3. Gestion des Clients

Les utilisateurs peuvent saisir facilement les informations d'un nouveau client, y compris le nom et le numéro de téléphone, ainsi que le montant de crédit associé.

En cliquant sur « Save » le client s'enregistre automatiquement dans la base de données et s'affiche sur le tableau.



The screenshot shows the 'Esalaf' web application interface. The sidebar on the left contains the Esalaf logo, a navigation menu with 'Home', 'Clients', 'Crédits', and 'Log out', and a 'Log out' button. The main content area is titled 'Gestion des clients' and features a table of clients and a form to add or update a client.

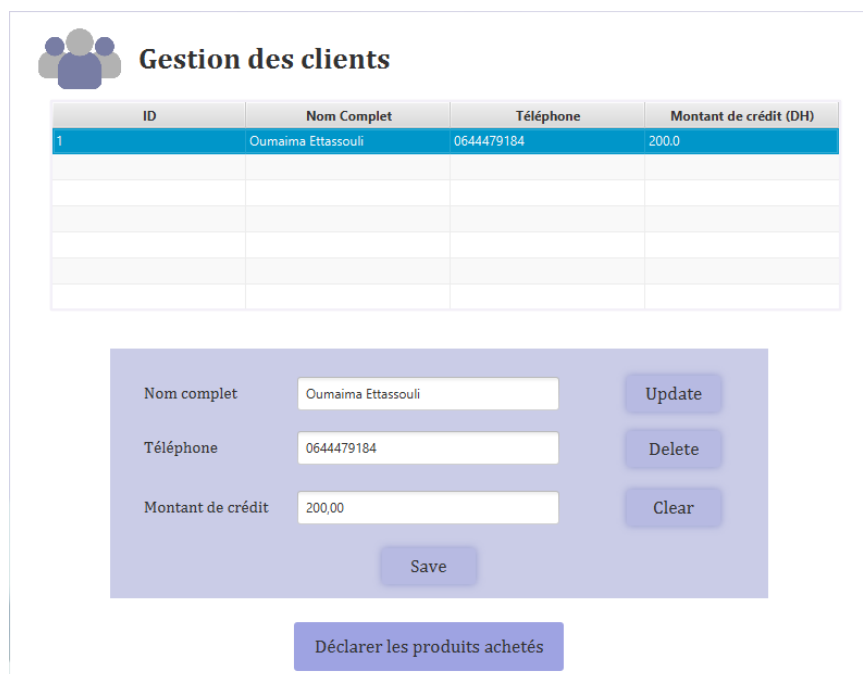
ID	Nom Complet	Téléphone	Montant de crédit (DH)
1	Oumaima Ettassouli	0644479184	200.0

The form below the table has the following fields and buttons:

- Nom complet:  Update
- Téléphone:  Delete
- Montant de crédit:  Clear
- Save

At the bottom of the form area is a button: Déclarer les produits achetés

L'utilisateur peut aussi modifier ou supprimer totalement les données d'un client de la liste en cliquant sur le client souhaité.



This screenshot shows the 'Gestion des clients' page with the client data populated in the form fields. The table shows the client 'Oumaima Ettassouli' with ID '1' and phone number '0644479184'.

ID	Nom Complet	Téléphone	Montant de crédit (DH)
1	Oumaima Ettassouli	0644479184	200.0

The form below the table has the following fields and buttons:

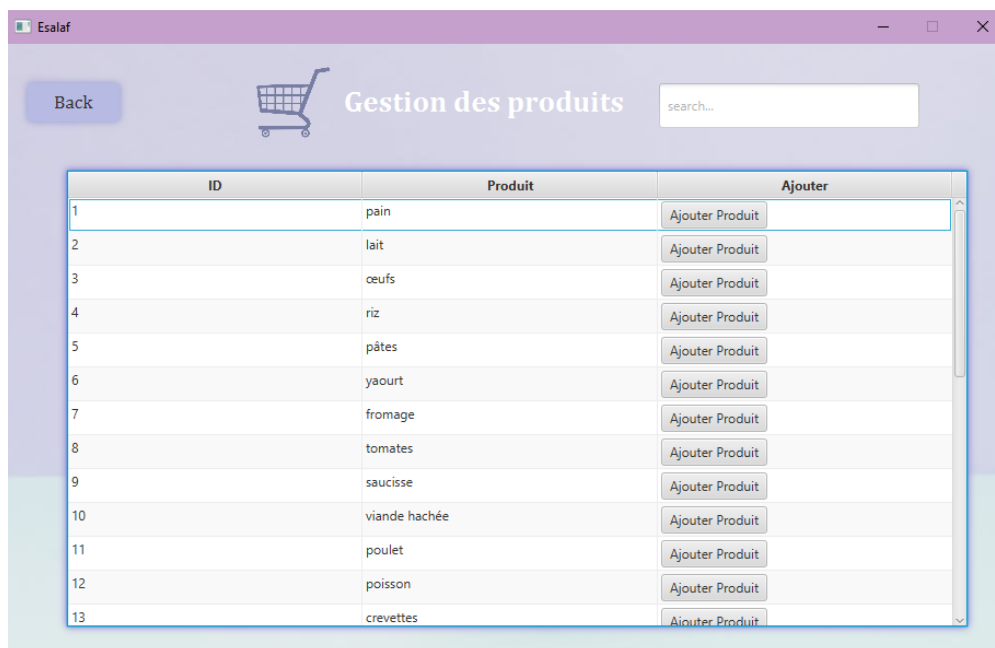
- Nom complet:  Update
- Téléphone:  Delete
- Montant de crédit:  Clear
- Save

At the bottom of the form area is a button: Déclarer les produits achetés



## 4. Déclaration des achats

Dans cette partie, l'utilisateur ajoute les différents produits achetés par le client. Il peut donc **chercher** le nom d'un produit en le saisissant dans « Search » ou le sélectionner directement de la liste déroulante.



### Search




Au moment où un produit est sélectionné, il s'enregistre automatiquement dans la table crédit du client.








## 5. Gestion des crédits

Ce tableau affiche la liste de tous les clients ayant un crédit, avec le montant reçu s'il n'a jamais reçu un montant d'un client, et qui par conséquent affiche le montant restant à payer une fois cliqué sur « Afficher le reste »



GÉREZ VOS CRÉDITS EN TOUTE SIMPLICITÉ

 Home
   
 Clients
   
 Crédits
   
 Log out



### Gestion des crédits

Afficher le reste

ID	Client	Reçu	Restant
1	Oumaima Ettassouli	0.0	0.0

### Gestion des crédits

Afficher le reste

	Client	Reçu	Restant
	Oumaima Ettassouli	50	0.0

#### ➤ Afficher le reste

	Reçu	Restant
	50.0	150.0

# Annexes

## Fonctions CRUD

### Save

```
1 //Fonction pour insérer les informations saisies dans DB
2 public void Saveclient() throws SQLException {
3     String sql = "INSERT INTO client "
4         + "(Nom, Telephone, Montant_credit) "
5         + "VALUES(?,?,?)";
6
7     DatabaseConnection connectNow = new DatabaseConnection();
8     Connection connectDB = connectNow.getConnection();
9
10    try {
11        Alert alert;
12        if (clientNom.getText().isEmpty()
13            || clientTel.getText().isEmpty()
14            || clientMontant.getText().isEmpty()){
15            alert = new Alert(Alert.AlertType.ERROR);
16            alert.setTitle("Error Message");
17            alert.setHeaderText(null);
18            alert.setContentText("Please fill all blank fields");
19            alert.showAndWait();
20        }
21        else {
22            prepare = connectDB.prepareStatement(sql);
23            prepare.setString(1, clientNom.getText());
24            prepare.setString(2, clientTel.getText());
25            prepare.setString(3, clientMontant.getText());
26            prepare.executeUpdate();
27
28            addClientsListData();
29            ClientShowListData();
30            Clear();
31        }
32    } catch (Exception e) {
33        e.printStackTrace();
34    }
35 }
36
37
```

## Update

```
1 public void Update() throws SQLException {
2
3     DatabaseConnection connectNow = new DatabaseConnection();
4     Connection connectDB = connectNow.getConnection();
5
6     String updateInfo = "UPDATE client SET Nom = '"
7         + clientNom.getText() + "', Telephone = '"
8         + clientTel.getText() + "', Montant_credit = '"
9         + clientMontant.getText()
10        + "' WHERE Nom = '"
11        + clientNom.getText() + "'";
12
13     prepare = connectDB.prepareStatement(updateInfo);
14     prepare.executeUpdate();
15
16     Alert alert = new Alert(Alert.AlertType.INFORMATION);
17     alert.setTitle("Information Message");
18     alert.setHeaderText(null);
19     alert.setContentText("Successfully Updated!");
20     alert.showAndWait();
21     Clear();
22     ClientShowListData();
23
24 }
```

## Delete

```
1 public void Delete() throws SQLException {
2
3     DatabaseConnection connectNow = new DatabaseConnection();
4     Connection connectDB = connectNow.getConnection();
5     String deleteInfo = "DELETE FROM client WHERE Nom = '"
6         + clientNom.getText() + "'";
7
8     prepare = connectDB.prepareStatement(deleteInfo);
9     prepare.executeUpdate();
10
11     Alert alert = new Alert(Alert.AlertType.INFORMATION);
12     alert.setTitle("Information Message");
13     alert.setHeaderText(null);
14     alert.setContentText("Successfully Deleted!");
15     alert.showAndWait();
16     Clear();
17     ClientShowListData();
18 }
```

## Search

```
1 public void ProductSearch() {
2     addProduitsList = addProduitsListData();
3
4     FilteredList<Produit> filter = new FilteredList<>(addProduitsList, e -> true);
5
6     search.textProperty().addListener((Observable, oldValue, newValue) -> {
7
8         filter.setPredicate(predicateProductData -> {
9
10             if (newValue == null || newValue.isEmpty()) {
11                 return true;
12             }
13
14             String searchKey = newValue.toLowerCase();
15
16             if (predicateProductData.getNom_produit().toLowerCase().contains(searchKey)) {
17                 return true;
18             } else {
19                 return false;
20             }
21         });
22     });
23     SortedList<Produit> sortList = new SortedList<>(filter);
24     sortList.comparatorProperty().bind(produitTableview.comparatorProperty());
25     produitTableview.setItems(sortList);
26 }
```

## Bibliographie

<https://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/cell/PropertyValueFactory.html#fromHistory>

<https://github.com/marcoman2/EmployeeManagementSystem.git>

<https://chat.openai.com>

<https://youtu.be/CGWRwpeihE8>

<https://youtu.be/pf7-zMCQllc>

<https://stackoverflow.com/questions/70308258/javafx-exception-in-application-start-method-java-lang-reflect-invoctantarget>

<https://levelup.gitconnected.com/solved-cant-load-java-fxml-file-from-different-package-346f41886c49>

<https://github.com/mahmoudhamwi/Fruits-Market.git>

---

# PETITE NOTE

## Remerciements

Je tiens à remercier mes professeurs pour l'opportunité  
Et l'idée de ce projet. Cela a été une expérience enrichissante  
Pour moi. Bien que le jeu ne soit pas parfait, j'ai apprécié  
chaque moment de ce travail.

| Oumaima Ettassouli