**Université Abdelmalek Essaadi Faculté ses Sciences et techniques de Tanger**

**Département Génie Informatique**

LST GI S6

**Programmation Orientée Objet en JAVA**

**Projet :** Esalaf

**Réalisé par :**

Zaoui Hanane

**Encadré par :**

**Pr. El Mokhtar EN-NAIMI**

**Pr. ELAACHAk LOTFI**

**Remerciement :**

Je tiens de prime abord à exprimer mes sincèresremerciements aux professeurs qui m'ont enseigné Java, car grâce à leur expertise, leur patience et leur dévouement, j'ai pu acquérir les compétences nécessaires pour créer cette application de crédit en JavaFx.

Leur enseignement m'a aidé à comprendre les concepts de programmation Java et à développer ma créativité en matière de développement d'applications.

Je suis vraiment reconnaissant pour tout ce que j'ai appris de vous et je tiens également à remercier toutes celles et ceux qui nous ont aidé de près ou de loin pour donner naissance à cet ouvrage. Merci encore pour votre contribution à mon éducation en Java.

**Table des matières**

[I. Introduction : 4](#_Toc131452535)

[II. Outils : 5](#_Toc131452536)

[III. Etapes création : 6](#_Toc131452537)

[1. Authentification : 6](#_Toc131452538)

[2. Home : 9](#_Toc131452539)

[3. Gestion des admins : 10](#_Toc131452540)

[4. Gestion des clients : 15](#_Toc131452541)

[5. Gestion de crédit : 17](#_Toc131452542)

[6. Gestion des paiements : 18](#_Toc131452543)

[IV. Conclusion : 21](#_Toc131452544)

# Introduction :

L'application développée est une application de crédit qui permet de gérer les clients, les utilisateurs, les crédits et les paiements. Elle a été développée en utilisant la plateforme JavaFX qui fournit une interface utilisateur graphique moderne et interactive.

L'objectif principal de cette application est de fournir une solution complète pour la gestion des crédits, depuis la création des clients jusqu'à la gestion des remboursements. Elle permet également de suivre les utilisateurs qui ont créé les crédits, les montants empruntés et les paiements effectués.

En outre, cette application fournit des fonctionnalités telles que la recherche de clients et de crédits, l'ajout de nouveaux clients, la création de nouveaux crédits et la gestion des paiements. Elle offre également une interface utilisateur intuitive et facile à utiliser pour une meilleure expérience utilisateur.

Dans ce rapport, nous allons décrire en détail les fonctionnalités de l'application et les technologies utilisées.

# Outils :

**JavaFX :** est une bibliothèque graphique pour Java permettant de créer des interfaces utilisateur riches et interactives pour les applications de bureau et mobiles. Elle offre de nombreuses fonctionnalités pour la conception d'interfaces, telles que des composants graphiques personnalisables, des animations, des effets visuels, des transitions.

**JDBC (Java Database Connectivity) :** est une API Java standard qui permet aux développeurs d'accéder aux bases de données relationnelles à partir de programmes Java. Elle fournit des classes et des interfaces pour établir une connexion à une base de données, exécuter des requêtes et des mises à jour, gérer les transactions.

**MySQL :** est un système de gestion de bases de données relationnelles open-source très populaire. Il est largement utilisé pour stocker des données pour les applications web et de bureau. MySQL prend en charge le langage SQL et offre des fonctionnalités telles que la gestion des transactions, les index, la sécurité.

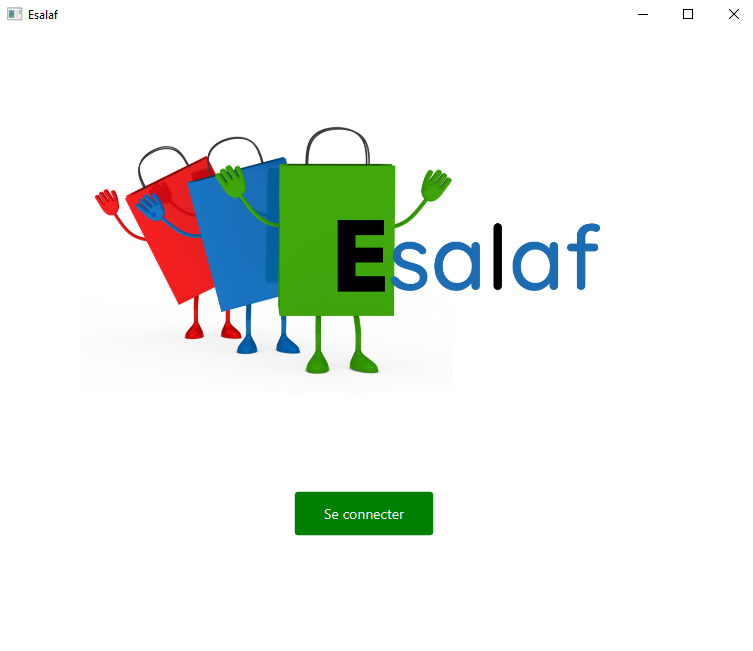
**Java :** est un langage de programmation orienté objet populaire et largement utilisé pour le développement d'applications de bureau, web et mobiles. Il offre de nombreuses fonctionnalités telles que la gestion de la mémoire, la gestion des exceptions, l'héritage, la polymorphie, etc. Java est également portable, ce qui signifie que les programmes Java peuvent être exécutés sur différents systèmes d'exploitation sans modification.

# Etapes création :

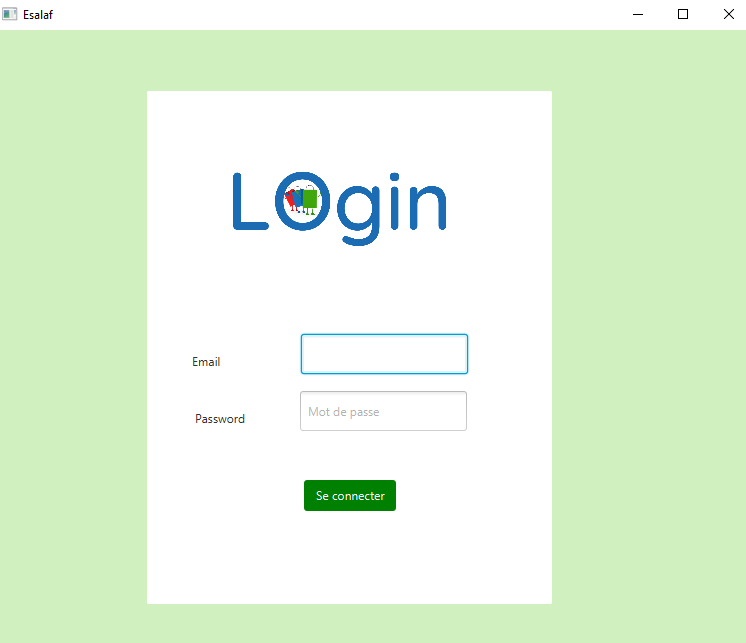
## Authentification :

Dans cette application, j'ai créé une interface ouverte qui contient le logo de l'application et un bouton pour s'authentifier. Cette interface est la première chose que l'utilisateur voit lorsqu'il ouvre l'application. En cliquant sur le bouton "se connecter", l'utilisateur est redirigé vers une page de connexion où il peut entrer son email et son mot de passe pour accéder aux fonctionnalités de l'application.

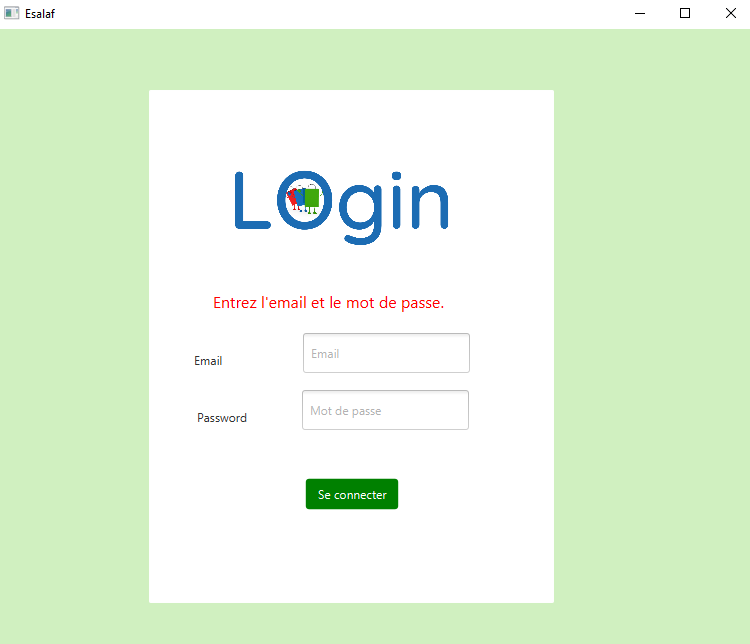
**Interface d’ouverte :**

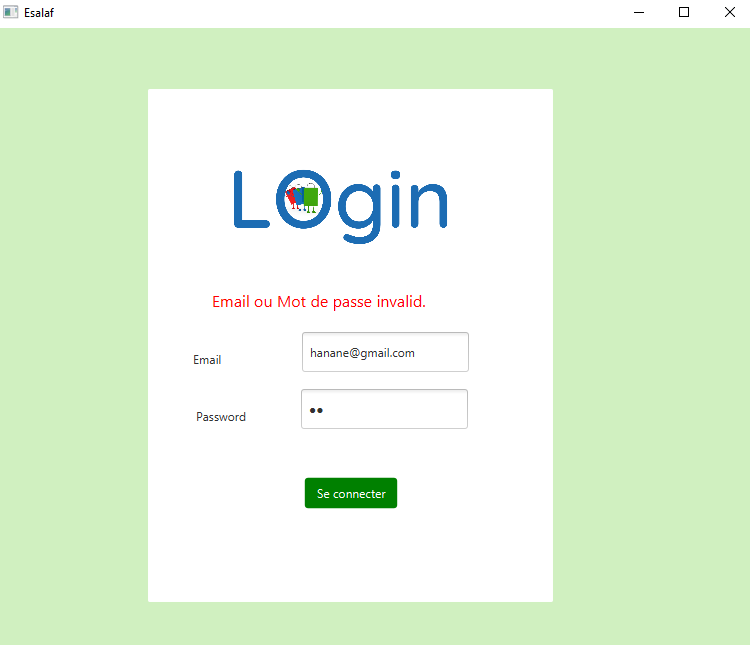


**Interface d’authentification :**



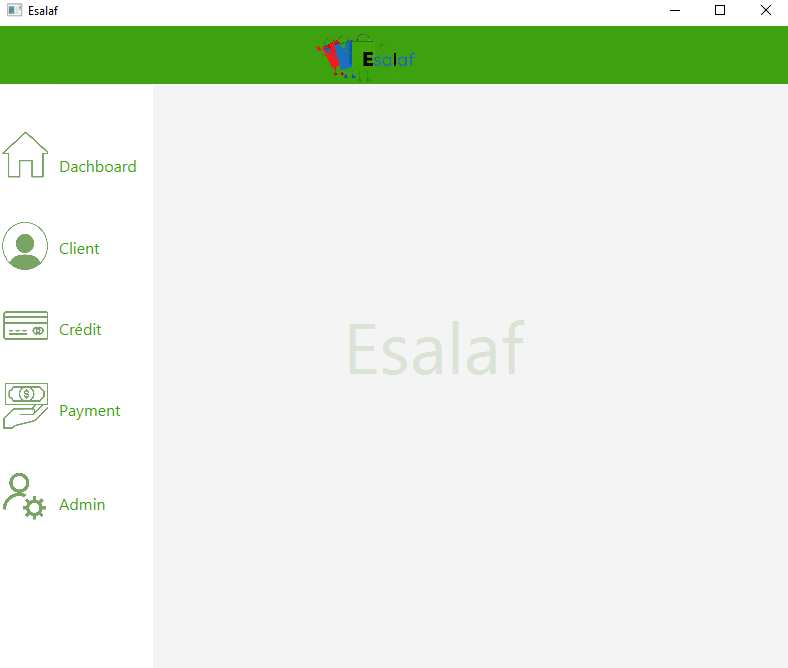
Dans cette interface j'ai mis en place un message d'erreur pour informer l'utilisateur s'il entre une ou plusieurs données incorrectes, comme une adresse e-mail ou un mot de passe incorrect. Ce message s’affichera pour aider l'utilisateur à corriger ses erreurs et à se connecter avec succès.





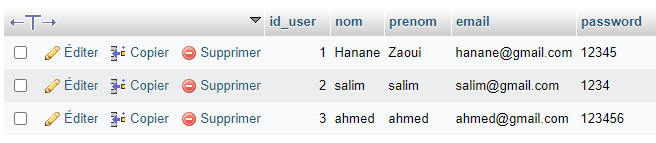
## Home :

Après l’authentification de l'utilisateur, l'interface Home s'affiche, elle contient un menu avec 5 éléments, chacun des éléments représente une gestion. Les cinq éléments du menu sont : "Gestion des clients", "Gestion des admins", "Gestion des crédits", "Gestion des paiements" et "Dachboard".



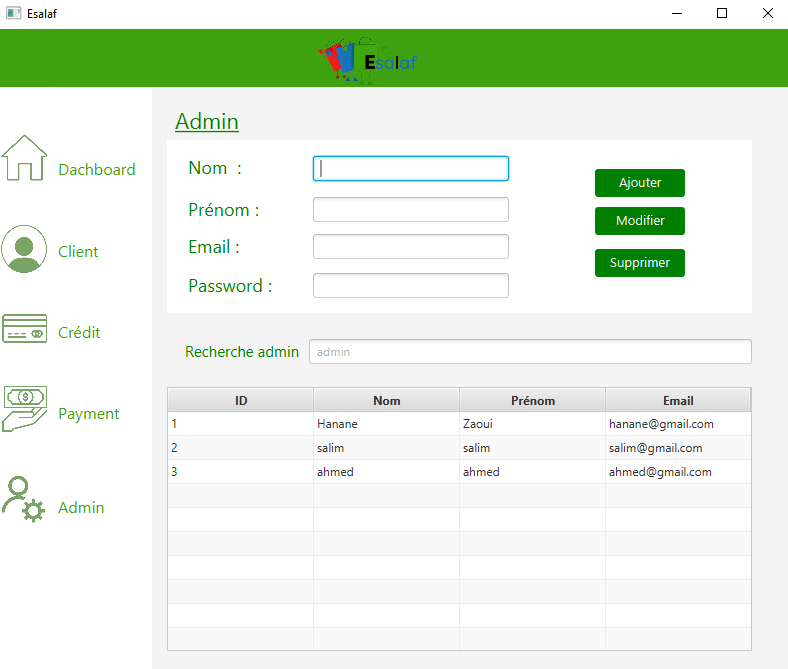
## Gestion des admins :

* **Tableau d’admin :**

****

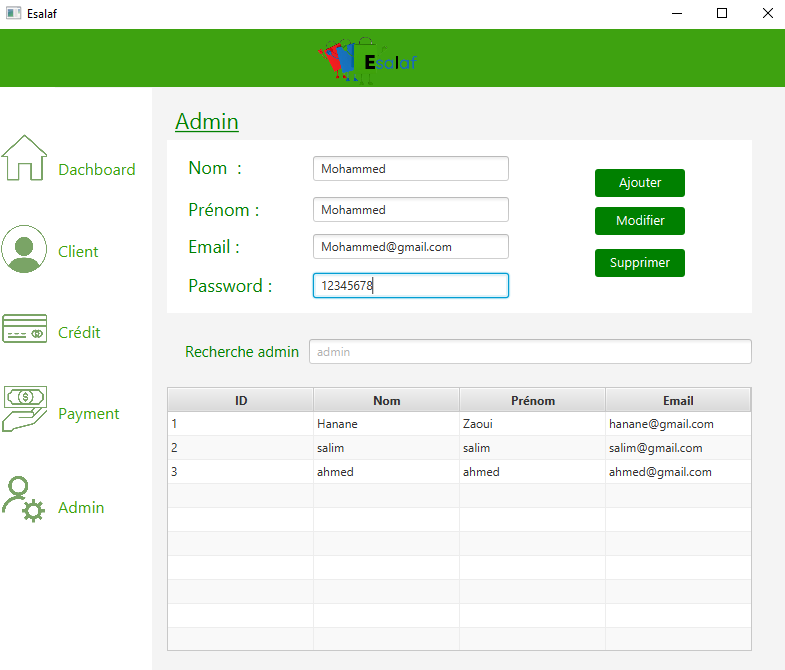
* **Interface des admins :**

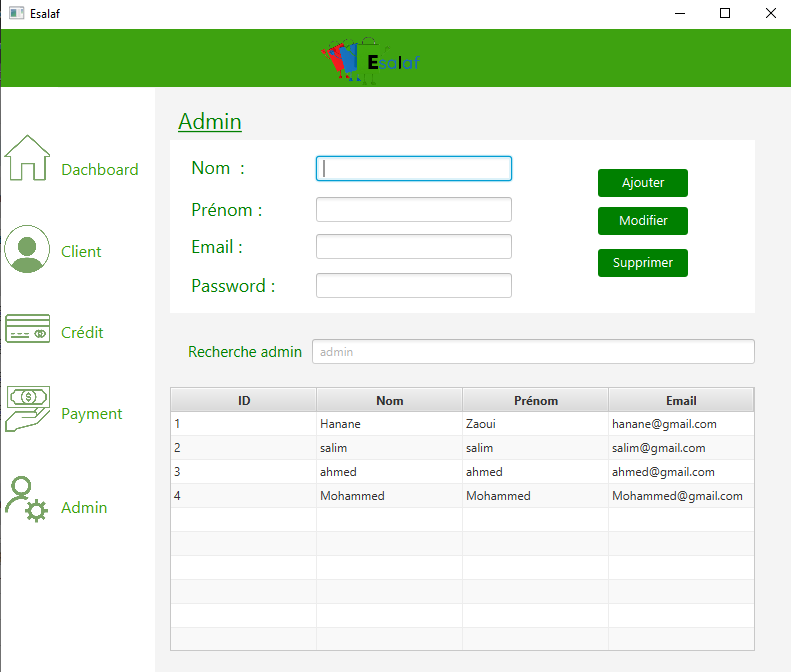
Dans l’interface de gestion des admins, nous pouvons afficher tous les admins dans un tableau, ajouter un admin, supprimer un admin et mettre à jour les informations d'un admin existant. Cela nous permet de gérer facilement les comptes d'administrateurs dans notre application.



* **Ajouter un admins**: j'ai utilisé une fonction appelée

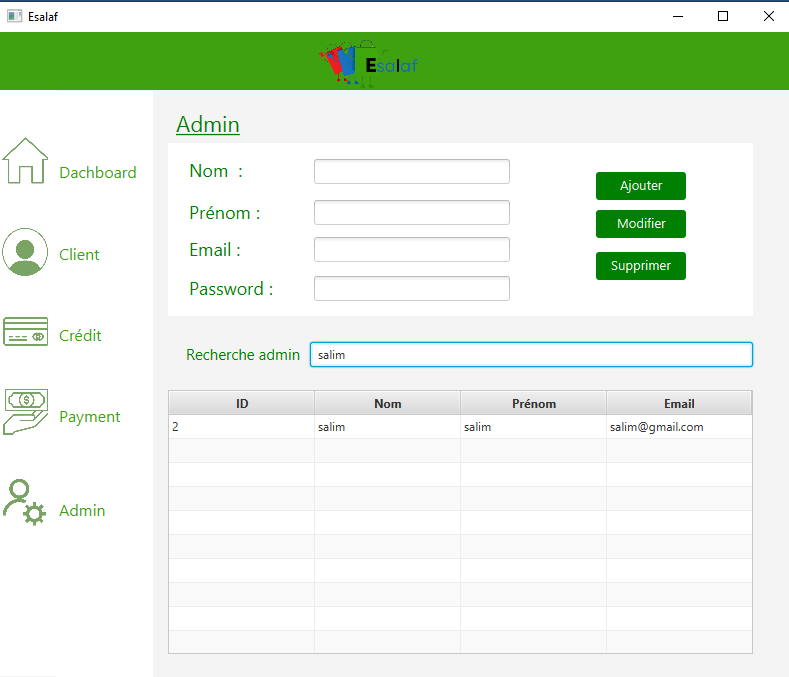
onSaveButtonClick(), qui prend les valeurs que nous écrivons dans les champs nom, prénom, email et mot de passe et les insère dans la base de données.

****

****

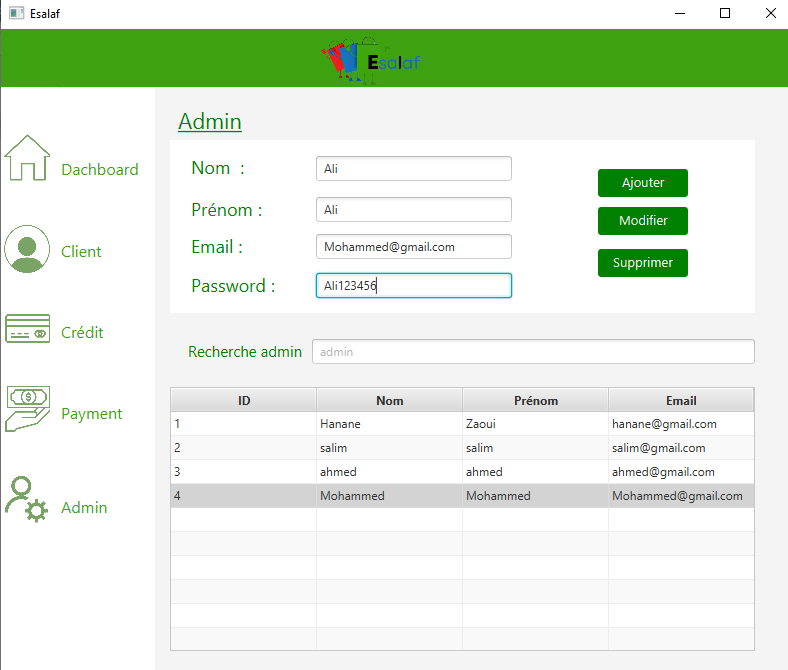
* **Rechercher admin**: j'ai utilisé une méthode pour trier

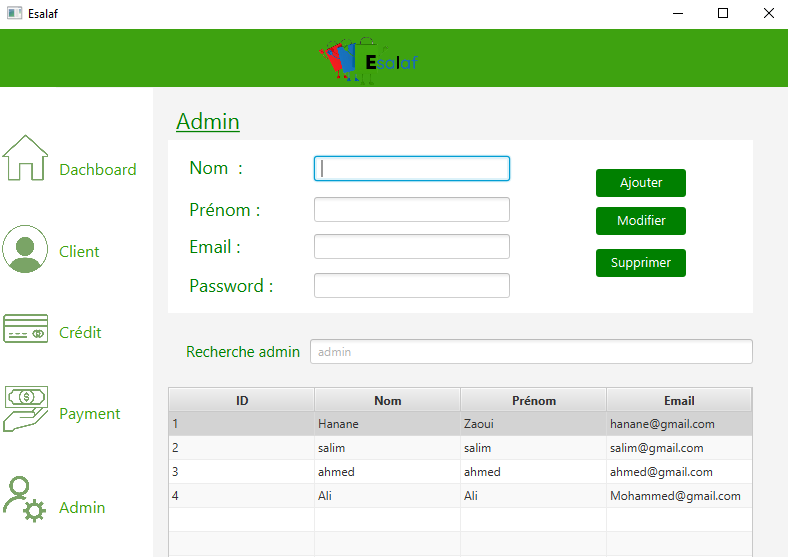
les données qui proviennent de la table des admins et je l'ai ajoutée à la fonction updateTable() qui affiche les données dans le tableau de vue.

****

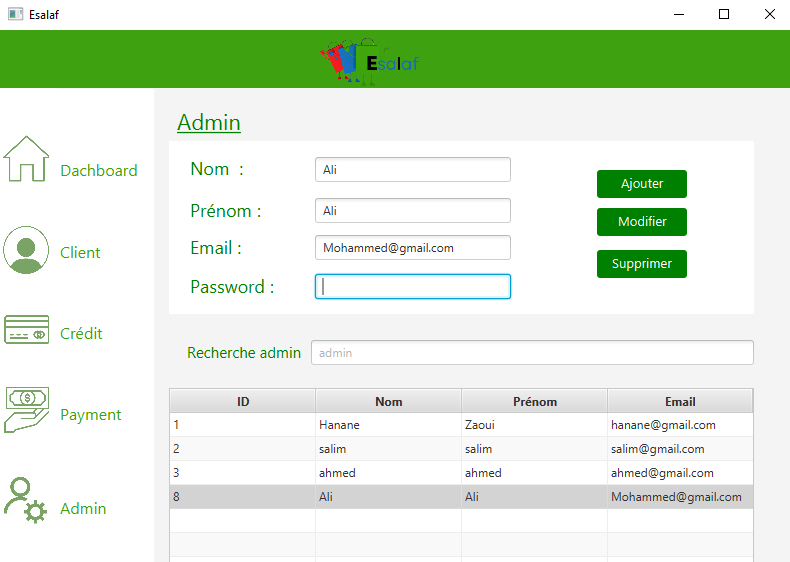
* **Modifier un admin**: j’ai utilisé deux fonctions. La

première, "setItems()", permet de sélectionner une ligne dans le tableau et de récupérer les valeurs correspondantes pour les afficher dans les champs "nom", "prenom", "email" et "mot de passe". La deuxième fonction onUpdateButtonClick() nous permet de mettre à jour les données dans la table de la base de données en récupérant les informations modifiées dans les champs.

****

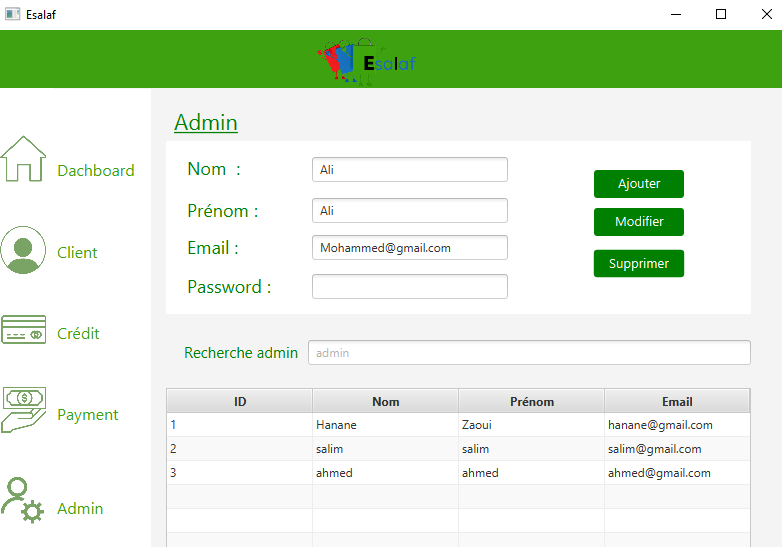
****

* **Supprimer un admin :**
* Sélectionner l’admin : en utilisant la fonction setItem()



* Cliquer sur le buton Supprimer : nous avons utilisé la fonction

"onDeleteButtonClick" qui envoie une requête à la base de données contenant les informations récupérées des champs "nom", "prénom", "email" et "mot de passe" pour supprimer l'élément correspondant.

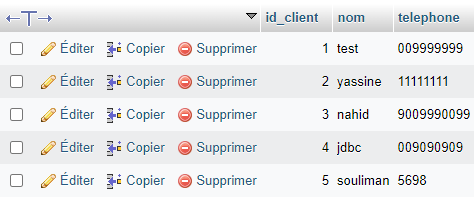


* **Elément du menu :** Dans le menu de l'application, j'ai

utilisé différentes fonctions pour passer d'une scène à une autre, telles que switchToClient(), switchToCredit() et switchToAdmin().

## Gestion des clients :

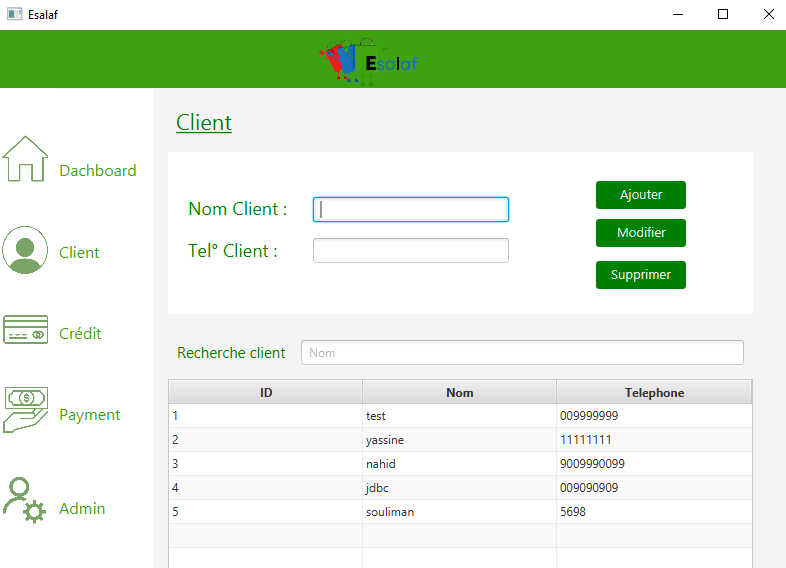
* **Table client :**



* **Interface des client :**

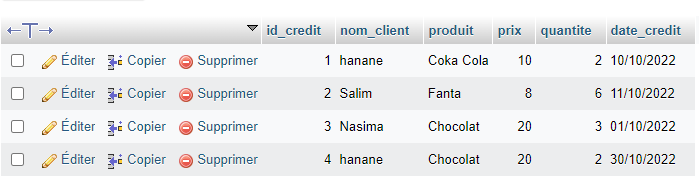
Dans l’interface de gestion des clients, nous pouvons afficher tous les clients dans un tableau, ajouter un client, supprimer un client et mettre à jour les informations d'un client existant. Cela nous permet de gérer facilement les comptes de nos clients dans notre application.

Dans cette interface, j’ai utilisé la même fonctionnalité que pour la gestion des administrateurs.



## Gestion de crédit :

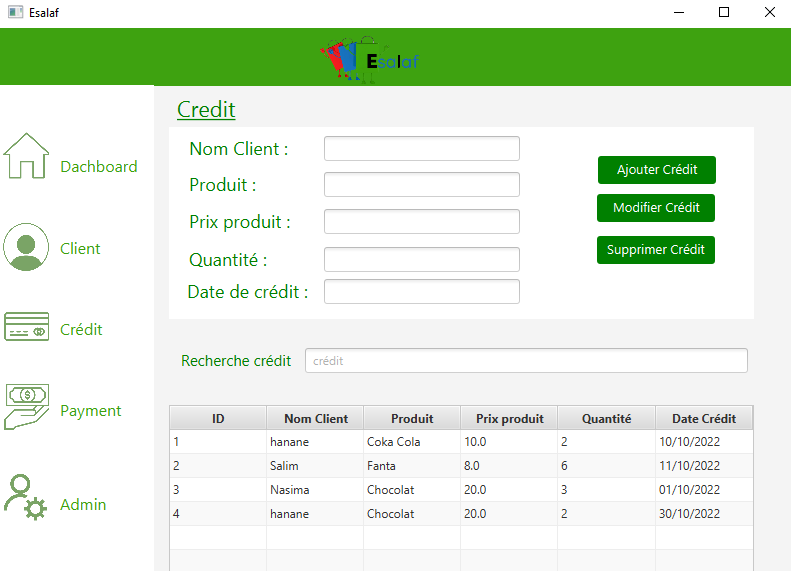
* **Tableau credit :**



* **Interface Credit :**

Dans l’interface de gestion des crédits, nous pouvons afficher tous les crédits dans un tableau, ajouter un crédit, supprimer un crédit et mettre à jour les informations d'un crédit existant. Cela nous permet de gérer facilement les crédits de nos clients dans notre application.

Dans cette interface, j’ai utilisé la même fonctionnalité que pour la gestion des administrateurs.



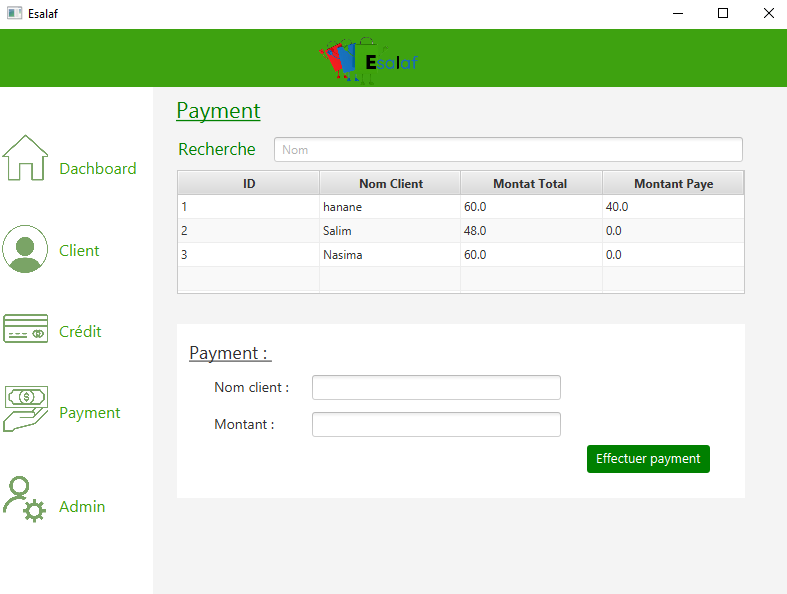
## Gestion des paiements :

* **Tableau de payment :**

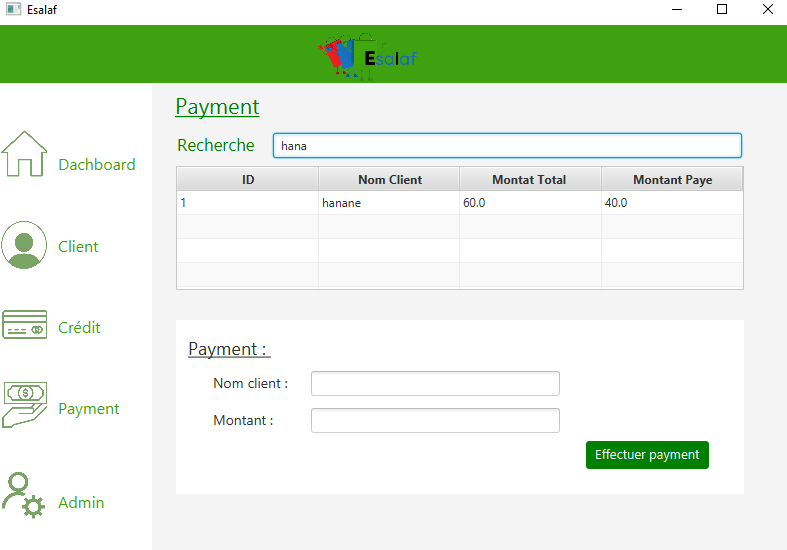


* **Interface de Paiement**: Dans l'interface de paiement,

j'ai mis en place une table pour afficher tous les paiements à effectuer et une barre de recherche pour rechercher un client spécifique. En outre, j'ai ajouté une zone de paiement pour permettre au client de payer son crédit**.**

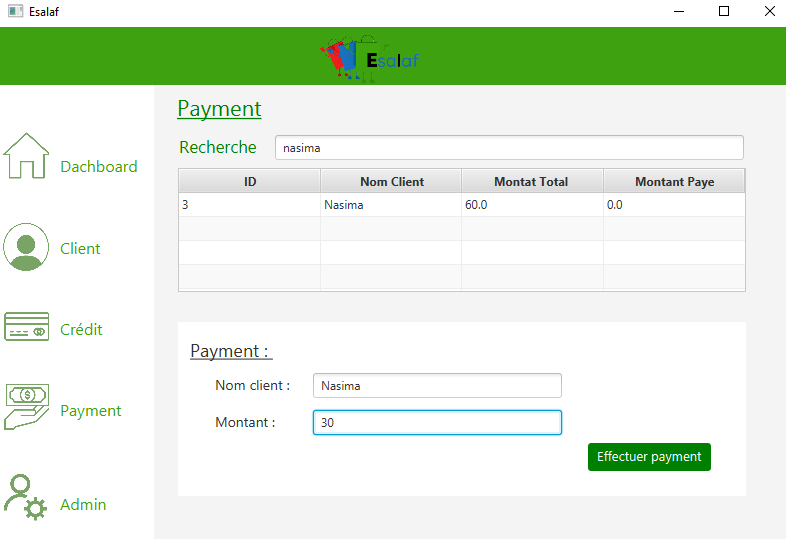


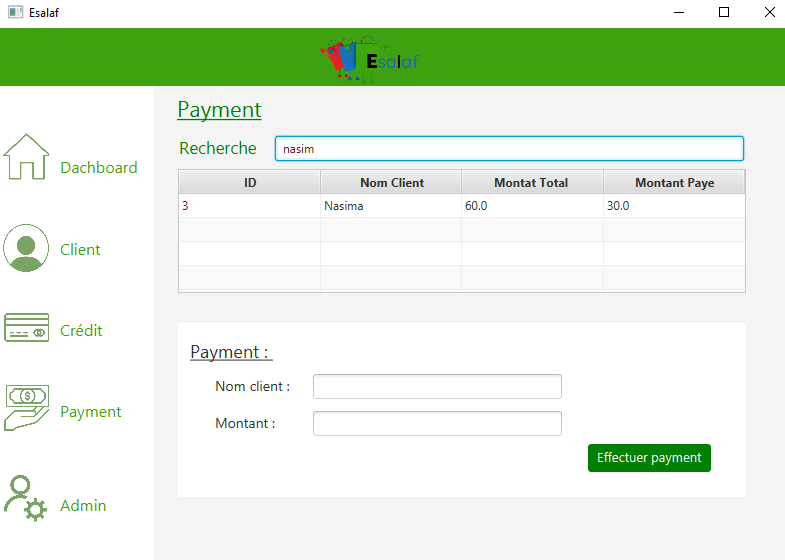
* **Recherche :**



* **Payment zone :** dans cette zone, nous pouvons

modifier le crédit du client en ajoutant le montant d'argent que le client fournit à l'administrateur dans le champ "Montant payé" et stocker cette information dans la base de données .





# Conclusion :

La gestion des prêts est une tâche complexe et fastidieuse lorsqu'elle est effectuée manuellement. C'est pourquoi la mise en place d'un système de gestion est nécessaire pour faciliter la tâche des agents et des clients. Grâce à la conception et à la réalisation de cette application de crédit en JavaFx, j'ai pu améliorer mon expérience et mes connaissances en programmation orientée objet en JAVA. Je suis fier du résultat obtenu et impatiente de poursuivre notre apprentissage dans ce domaine passionnant.