**Esalaf**

**(Project)**

Encadré par :

*Prof.* El Mokhtar EN-NAIMI

*Réalisé par :*

*Chibani Fahd*

# SOMMAIRE

[SOMMAIRE 2](#_bookmark0)

[INTRODUCTION 3](#_bookmark1)

[LE PROCESSUS DU DÉVELOPPEMENT ET LESOPTIONS DÉVELOPPÉES 4](#_bookmark2)

[1. Login interface 4](#_bookmark3)

1. [Main interface 5](#_bookmark4)
2. [Interface Client 5](#_bookmark5)
3. [Interface Produit 6](#_bookmark6)
4. [Interface Crédit 6](#_bookmark7)
5. [Interface Commande 7](#_bookmark8)

[Les fichier de travail 7](#_bookmark9)

* + [Client.java 8](#_bookmark10)
  + [ClientController 9](#_bookmark11)

# INTRODUCTION

L’objectif principal de ce projet est de maitrisé les interfaces graphique et la programmation orientée objet en java par la miseen place d‘une d'une application desktop java , basée sur JDBC et JavaFX, l'application doit

gérer les crédit, les commandes , les produits, les clients, ainsi elle donne un tableau de bord générique a l’aide de IDE (Integrated Development Environnent ) IntelliJ IDEA qu’il est un environnement de développement intégré destiné au développement de logiciels informatiques reposant sur la technologie Java. .

L’application desktop java développé s’appelle Esalaf. Cette application se constitue de six interface :

* Login Interface
* Main Interface
* Interface Client
* Interface Crédit
* Interface Produit
* Interface Commande

Outils : Intellij idea, JavaFX, JDBC, MySQL, Java, SceneBuilder.

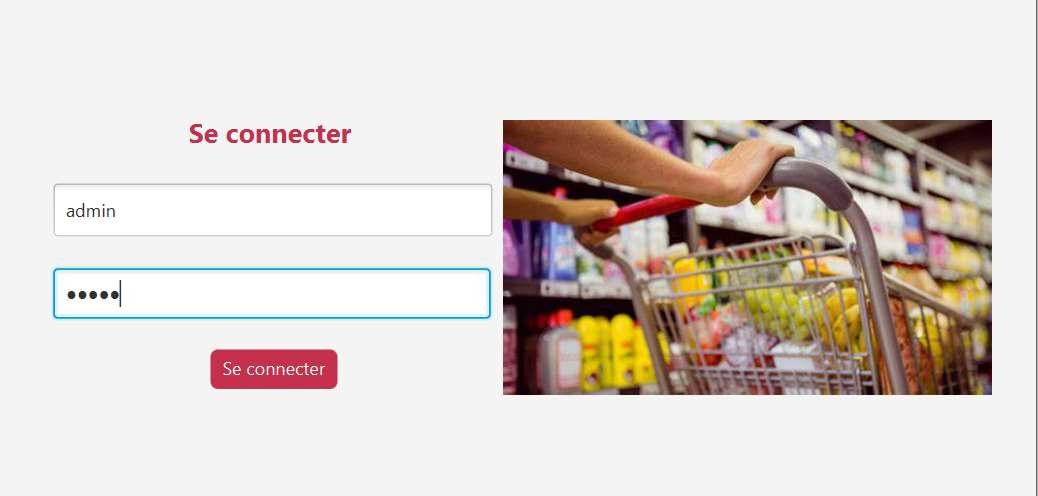
# LE PROCESSUS DU DÉVELOPPEMENT ET LES OPTIONS DÉVELOPPÉES

Afin de développer notre application desktop, on avait besoin de créer plusieurs Interface comme déja indiquer, qui sont les différents interfaces visualiser par

l’utilisateur et qui le permettre de stocker les différents information (Client, Crédit, Produit, Commande), et ajouter ensuite des méthodes permettant la navigation d’une interface vers une autre.

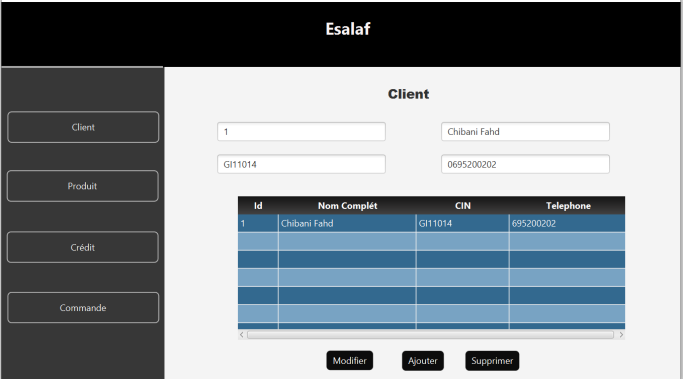
Les interfaces créées sont les suivants :

### 1. Login interface :



1. **Main interface :**

### Interface Client :

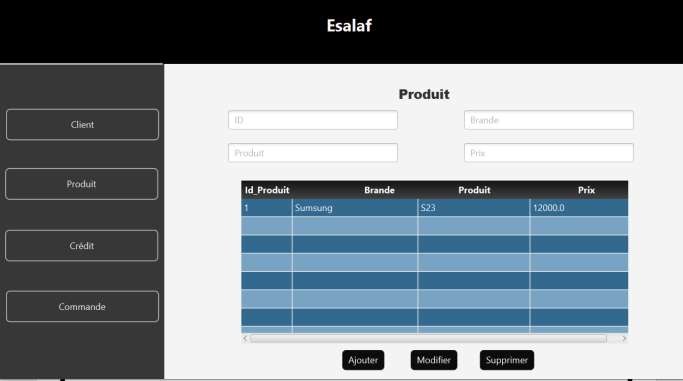


LST Geni\_Info

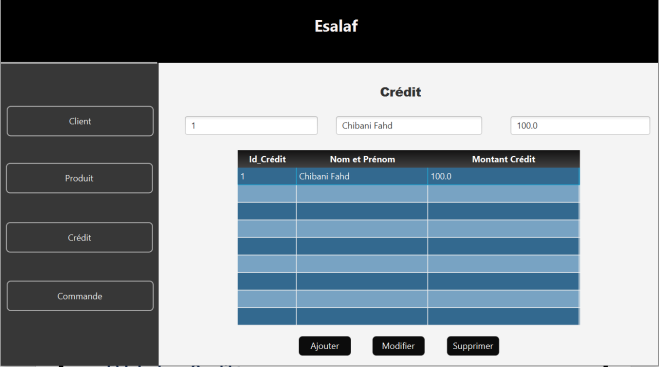
Chibani Fahd Grp1

2022-2023

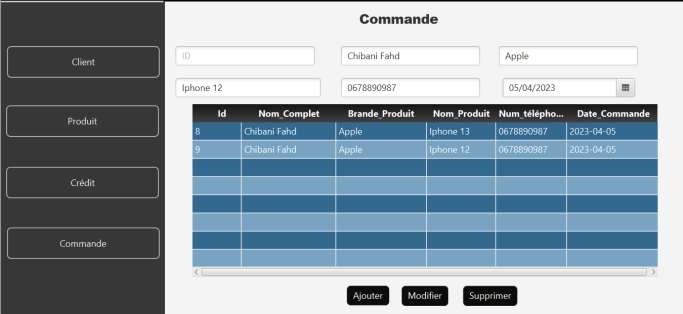
### Interface Produit :



### Interface Crédit :

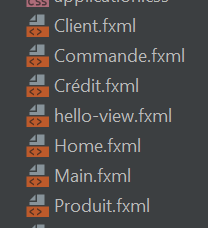


### Interface Commande :



**Les fichier de travail :**

Chaque Interface se constitue de deux classe, une classe contient le constructeur pour initialiser les attributs d’interface et une classe Controller qui gère la connexion avec la base donne et les méthode CRUD est un fichier.fxml qui contient le design de chaque interfaces .



Si en prend l’interface Client par exemple :

#### Client.java :

* public class Client { private int IdC ;

private String NomprenomC ; private String Cni ; private String Telephone ;

public Client(int IdC, String NomprenomC, String Cni,String Telephone) {

this.IdC = IdC; this.NomprenomC= NomprenomC; this.Telephone = Telephone; this.Cni = Cni;

}

public int getIdC() { return IdC;

}

public void setidC(int IdC) { this.IdC = IdC;

}

public String getNomprenomC() { return NomprenomC;

}

public void setNomprenomC(String NomprenomC) { this.NomprenomC = NomprenomC;

}

public String getTelephone() { return Telephone;

}

public void setTelephone(String Telephone) { this.Telephone = Telephone;

}

public String getCni() { return Cni;

}

public void setCni(String Cni) { this.Cni = Cni;

}

}

* + **ClientController :**
* Les méthodes CRUD :
* public ObservableList<Client> getclientsListe(){ ObservableList<Client> ClientsList =

FXCollections.*observableArrayList*(); Connection conn = getConnection(); String req = "SELECT \* FROM client"; Statement st;

ResultSet rs;

try{

st = conn.createStatement(); rs = st.executeQuery(req); Client clients; while(rs.next()){

clients = new Client(rs.getInt("IdC")

,rs.getString("NomprenomC")

,rs.getString("Cni")

,rs.getString("Telephone")); ClientsList.add(clients);

}

}catch (Exception ex){ ex.printStackTrace();

}

return ClientsList;

}

public void showclients(){

ObservableList<Client> Liste = getclientsListe(); colid.setCellValueFactory(new

PropertyValueFactory<>("IdC")); colnom.setCellValueFactory(new

PropertyValueFactory<Client,String>("NomprenomC")); colcni.setCellValueFactory(new

PropertyValueFactory<Client,String>("Cni")); coltel.setCellValueFactory(new

PropertyValueFactory<Client,String>("Telephone")); tvBox.setItems(Liste);

}

public void onAjouterButtClick(){ String c="NULL";

String req ="INSERT INTO client VALUES ("+ c

+",'"+tfnom.getText()+"','"+tfcni.getText()+"',"+tfnum.getTe xt()+")";

executeQuery(req);

showclients();

}

public void onmodButtClick(){

String req = "UPDATE client SET NomprenomC = '" + tfnom.getText() + "', Cni = '" + tfcni.getText()

+ "', Telephone= " + tfnum.getText() + " WHERE idC

= " + tfid.getText() + ""; executeQuery(req); showclients();

}

public void onsuppButtClick() {

String req = "DELETE FROM client WHERE IdC =" + tfid.getText() + "";

executeQuery(req); showclients();

}

public void executeQuery(String req) { Connection conn = getConnection(); Statement st;

try{

st = conn.createStatement(); st.executeUpdate(req);

}catch(Exception ex){ ex.printStackTrace();

}

}

C’est méthode CRUD nous aides a afficher, ajouter, modifier et supprimer les élément des tableaux.

* ***Méthode de connexion au data base :***

- public Connection getConnection(){ Connection conn;

try { conn =

DriverManager.*getConnection*("jdbc:mysql://127.0.0.1:3306/esa laf","root","");

return conn;

}catch (Exception ex){

System.*out*.println("Error "+ ex.getMessage()); return null;

}

}