

RAPPORT DE ESALAF PROJET



Realiser par:

Marouan Daghmoumi

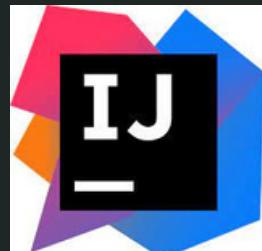
Encadrer par:

Pr. El Mokhtar EN-NAIMI
Pr.Oussama Ndama
Pr.Iotfi El Aachak

Les Outils

INTELLIJ:

logiciel pour le développement de logiciels



JAVAFX:

pou développement d'interface graphique (GUI)



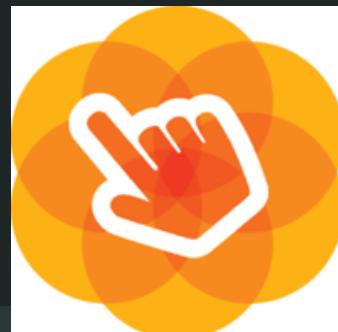
MYSQL:

comme base donnee

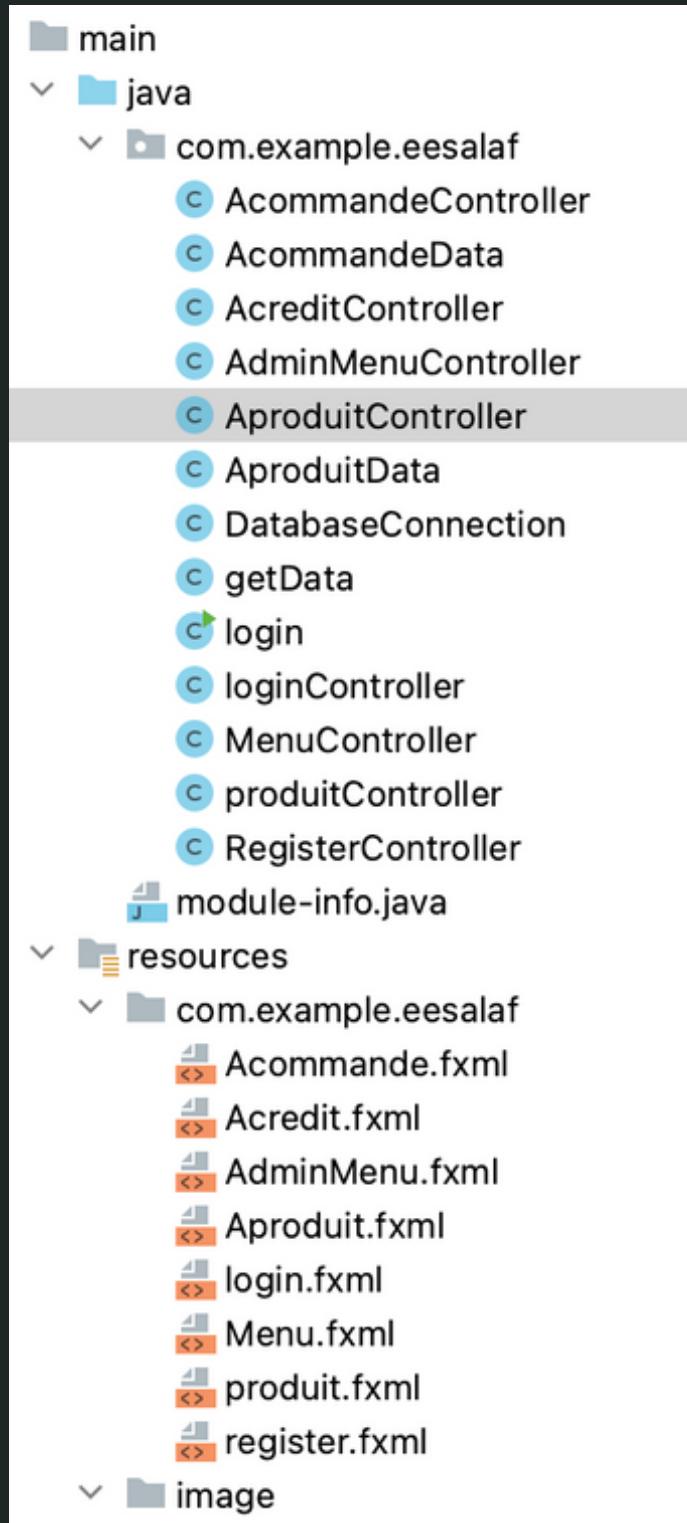


SCENE BUILDER:

design les interface



Les fichiers



les fichiers Controller :

Contien les classes de chaque page ,leurs actions et la declaration des variable .

les fichiers FXML :

Contien design de chaque interface .

les fichiers Data :

Contien la connection avec data base .

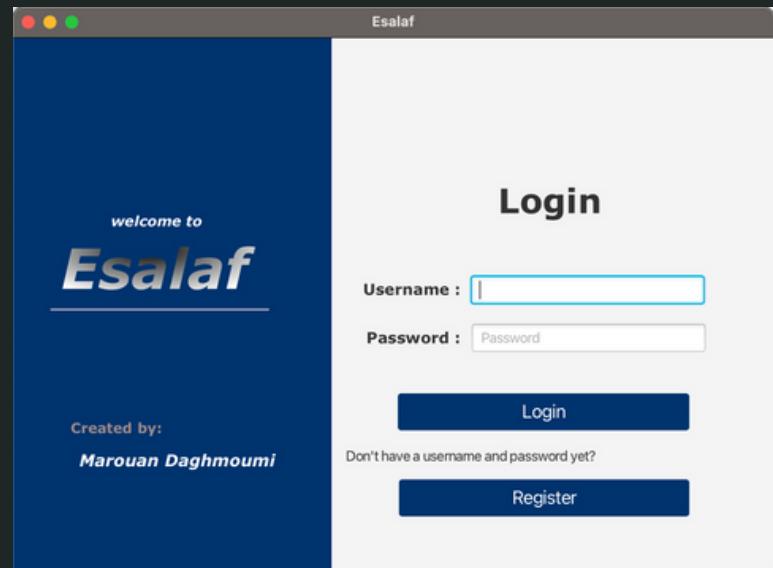
Dossier image :

Contien la connection avec data base .

Login Page

fichier FXML : login.FXML

fichier Controller : loginController.java



On declare les button ,les textField,
les labels et passwordField
qui ce trouve dans l design

```
1 usage
public class loginController {
    2 usages
    @FXML
    private Button cancelButton;
    4 usages
    @FXML
    private Label loginMessageLabel;
    5 usages
    @FXML
    private TextField usernameTextField;
    4 usages
    @FXML
    private PasswordField passwordPasswordField;
```

fonction qui ne permet
de dire a utilisateur de
saisir les donners

```
public void loginButtonOnAction(ActionEvent e){
    getData.username=usernameTextField.getText();
    if (!usernameTextField.getText().isBlank() && !passwordPasswordField.getText().isBlank()){
        loginMessageLabel.setText("bienvenue");
        validateLogin();
        validateLogin1();

    }else {
        loginMessageLabel.setText("Enter your username and password");
    }
}
```

fonction qui ne permet de vérifier si utilisateur existe dans notre base données , si non on l'affiche message d'error .

```

public void validateLogin(){
    DatabaseConnection connectNow = new DatabaseConnection();
    Connection connectDB = connectNow.getConnection();

    String verifyLogin = "SELECT count(1) FROM client WHERE username = '" + usernameTextField.getText() + "' AND password = '" + passwordTextField.getText() + "'";

    try {
        Statement statement = connectDB.createStatement();
        ResultSet queryResult = statement.executeQuery(verifyLogin);

        while (queryResult.next()){
            if (queryResult.getInt("columnIndex_1") == 1){
                createAccountForm();
            } else{
                loginMessageLabel.setText("Username or password incorrect");
            }
        }
    } catch (Exception e){
        e.printStackTrace();
    }
}

```

la fonction qui cree la page de register (enscrire dans notre site), la meme fonction nous permet de cree la page home de admin et user.

```

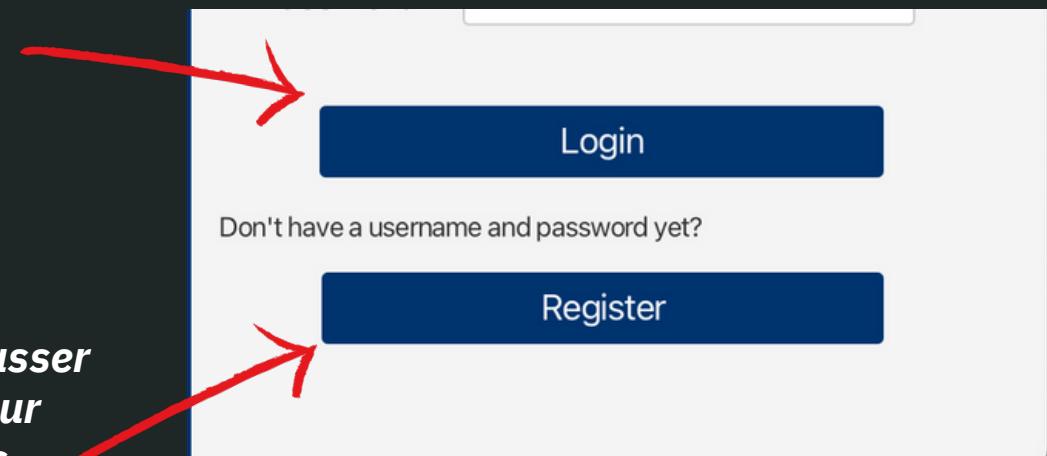
1 usage
public void createAccountForm(){
try {

    FXMLLoader fxmlLoader = new FXMLLoader(login.class.getResource( name: "register.fxml"));
    Stage registerstage =new Stage();
    Scene scene = new Scene(fxmlLoader.load(), v: 700, v1: 500);
    registerstage.setTitle("Esalaf");
    registerstage.setScene(scene);
    registerstage.show();

} catch (Exception e){
    e.printStackTrace();
    e.getCause();
}
}

```

login button pour passer a 'interface home soit d'admin ou user apres verification de la base donnees

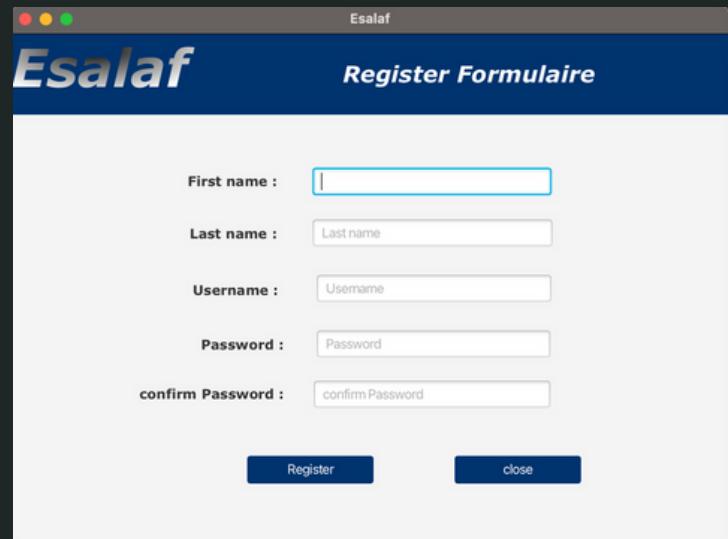


register button pour passer a l'interface register pour inscrire dans la basees donnees

Register Page

fichier FXML : Register.FXML

fichier Controller : registerController.java



On declare les button ,....

```
mysql> SELECT * FROM client;
+----+-----+-----+-----+-----+
| idclient | Firstname | Lastname | Username | Password |
+----+-----+-----+-----+-----+
| 1 | Marouan | Daghmoui | Moro231 | mama
| 2 | Ahmad | hasani | ahmd1 | hasan
| 3 | hamid | yusufi | yuu1 | hamiid
| 4 | malak | Daghmoui | Mimi | 1111
| 5 | Iyad | Daghmoui | speed1 | Iyad.2002
| 6 | Loubna | Senhaji | Loubna12 | lolo
| 7 | hgnd | jgc | A | A
| 8 | mnana | ouazani | mina | 111
| 9 | marouan | Daghmoui | Marouan | 1111
+----+-----+-----+-----+-----+
9 rows in set (0.05 sec)

mysql> SELECT * FROM admin;
+----+-----+-----+-----+-----+
| idadmin | Firstname | Lastname | Username | Password |
+----+-----+-----+-----+-----+
| 1 | Marouan | Daghmoui | Moro231 | mama
| 2 | Marouan | Daghmoui | moro | 111
+----+-----+-----+-----+
2 rows in set (0.00 sec)
```

les tableau de data bases .
on a deux tableaux tableaux client et
admin

Pour enregistrer les
donnees saiser a la bases
donnees

```
2 usages
@FXML
private Button closeButton;
2 usages
@FXML
private Label registrationMessageLabel;
3 usages
@FXML
private TextField setPasswordField;
2 usages
```

```
public void registerUser(){
    DatabaseConnection connectNow = new DatabaseConnection();
    Connection connectDB = connectNow.getConnection();

    String firstname = firstnameTextField.getText();
    String lastname = lastnameTextField.getText();
    String username = usernameTextField.getText();
    String password = setPasswordField.getText();

    String insertFields = "INSERT INTO client(firstname,lastname,username,password) VALUES('";
    String insertValues = firstname + "','" + lastname + "','" + username + "','" + password + "')";
    String insertToRegister = insertFields + insertValues;

    try{
        Statement statement = connectDB.createStatement();
        statement.executeUpdate(insertToRegister);

        registrationMessageLabel.setText("User has been registered successfully");
    }catch (Exception e){
        e.printStackTrace();
        e.getCause();
    }
}
```

partie Admin

Button initialiser la page

Button pour passer a l'interface produit

Button pour passer a l'interface credit

le nom d' Admin



la fonction qui nous permet de switcher entre les interfaces

```
1 usage
@FXML
private void home(MouseEvent event) { bp.setCenter(ap); }

1 usage
@FXML
private void produit(MouseEvent event) { loadPage("Aproduit"); }

1 usage
@FXML
private void credit(MouseEvent event) { loadPage("Acredit"); }

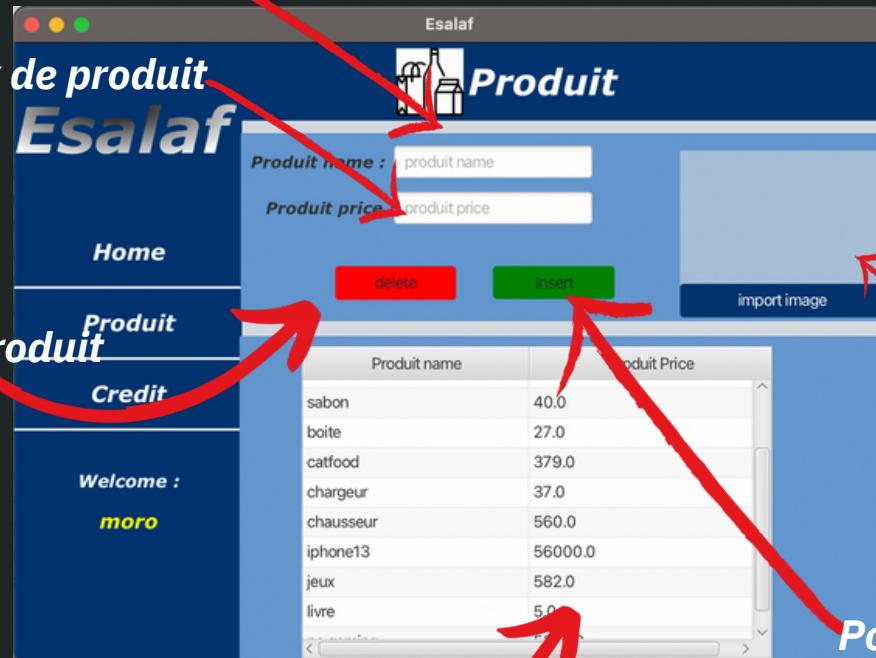
no usages
@FXML
private void commande(MouseEvent event) { loadPage("Acommande"); }

3 usages
private void loadPage(String page){
    Parent root = null;

    try {
        root= FXMLLoader.load(getClass().getResource( name: page+".fxml"));
    } catch (IOException e) {
        Logger.getLogger(AdminController.class.getName()).log(Level.SEVERE, msg: null,e);

    }
    bp.setCenter(root);
}
```

Pour saisir le nom de produit



Pour saisir le prix de produit

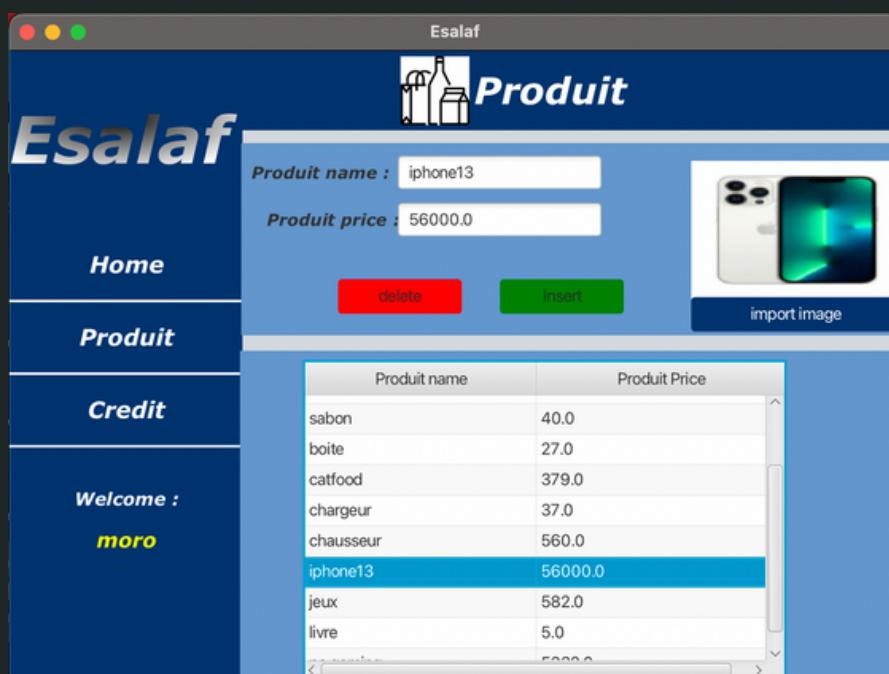
pour importer les images

Pour supprimer produit

Pour insert produit

Tableau des produits existe

Si on a selectioner un produit de tableaux



il nous a feches les inforamtion qui sont existe dans bases donnees , on peut le enregister ,modifier,ou supprimer .

fondction qui nous permet de supprimer produit selectionner .

```

public void Delete() {
    String sql = "DELETE FROM produit WHERE produitname = '" + Produitname.getText() + "'";
    connect = DatabaseConnection.getConnection();

    try {
        Alert alert;
        if (Produitname.getText().isEmpty() || Produitprice.getText().isEmpty() || getData.path == null || getData.path == "") {
            alert = new Alert(Alert.AlertType.ERROR);
            alert.setTitle("Error message");
            alert.setHeaderText(null);
            alert.setContentText("Please fill all blank fields");
            alert.showAndWait();
        } else {
            alert = new Alert(Alert.AlertType.CONFIRMATION);
            alert.setTitle("Confirmation Message");
            alert.setHeaderText(null);
            alert.setContentText("Are you sure you want to DELETE produit: " + Produitname.getText() + "?");
            Optional<ButtonType> option = alert.showAndWait();

            if (option.get().equals(ButtonType.OK)) {
                statement = connect.createStatement();
                statement.executeUpdate(sql);

                alert = new Alert(Alert.AlertType.INFORMATION);
                alert.setTitle("Information Message");
                alert.setHeaderText(null);
            }
        }
    } catch (Exception e) {
        System.out.println(e);
    }
}

```

idproduit	produitname	produitprice	image
13	fdsbé'rér	133	/Users/marouandgh/Desktop/GI.png
18	coscos	20	/Users/marouandgh/Desktop/eesalaf/src/main/resources/image/coscos.jpeg
19	zit	25	/Users/marouandgh/Desktop/eesalaf/src/main/resources/image/oil.jpeg
20	sabon	40	/Users/marouandgh/Desktop/eesalaf/src/main/resources/image/savon.jpeg
22	boite	27	/Users/marouandgh/Desktop/eesalaf/src/main/resources/image/boite.jpeg
23	catfood	379	/Users/marouandgh/Desktop/eesalaf/src/main/resources/image/catfood.jpeg
24	chargeur	37	/Users/marouandgh/Desktop/eesalaf/src/main/resources/image/chargeur.jpeg
25	chausseur	560	/Users/marouandgh/Desktop/eesalaf/src/main/resources/image/chaussure.jpeg
26	iphone13	56000	/Users/marouandgh/Desktop/eesalaf/src/main/resources/image/iphone.jpeg
27	jeux	582	/Users/marouandgh/Desktop/eesalaf/src/main/resources/image/jeux.jpeg
28	livre	5	/Users/marouandgh/Desktop/eesalaf/src/main/resources/image/livre.jpeg
29	pc gaming	5332	/Users/marouandgh/Desktop/eesalaf/src/main/resources/image/pc.jpeg

Notre data base contien 4 colons idproduit,produitname,produitprice et image importer

fondction qui nous permet de inserer produit apres saiser les donneesit

```

public void produitAdd() {
    String sql = "INSERT INTO produit (produitname, produitprice, image) VALUES(?, ?, ?) ";
    connect = DatabaseConnection.getConnection();

    try {
        Alert alert;
        if (Produitname.getText().isEmpty() || Produitprice.getText().isEmpty() || getData.path == null || getData.path == "") {
            alert = new Alert(Alert.AlertType.ERROR);
            alert.setTitle("Error message");
            alert.setHeaderText(null);
            alert.setContentText("Please fill all blank fields");
            alert.showAndWait();
        } else {
            prepare = connect.prepareStatement(sql);
            prepare.setString(parameterIndex: 1, Produitname.getText());
            prepare.setString(parameterIndex: 2, Produitprice.getText());

            String uri = getData.path;
            uri = uri.replace(target: "\\", replacement: "\\\\");

            prepare.setString(parameterIndex: 3, uri);
            prepare.executeUpdate();

            alert = new Alert(Alert.AlertType.INFORMATION);
            alert.setTitle("Information Message");
            alert.setHeaderText(null);
        }
    } catch (Exception e) {
        System.out.println(e);
    }
}

```

The screenshot shows the Esalaf application interface. On the left, there is a sidebar with navigation links: Home, Produit, Credit, Welcome : moro. The main content area displays a table titled "Tableau des produits acheter" (List of purchased products) with columns: nom, produit nom, and produit price. The data in the table is as follows:

	nom	produit nom	produit price
		batata	12.0
		fdsb	1334.0
A		fdsb	1334.0
marouan		batata	12.0
A		fdsb	1334.0
A		fdsb	1334.0
A		batbot	12.0

At the bottom of the main content area is a green button labeled "voir les edits". A red arrow points from the text "Tableau des produits acheter" to this button.

Tableau des produits acheter

Dans Interface credit ,Admin peut avoir tout les produits qui sont acheter par les users

Notre data base credit

```
mysql> SELECT * FROM credit;
+-----+-----+-----+-----+
| idcredit | username | produitname | produitprice |
+-----+-----+-----+-----+
|      1 |         | batata     |    12.0
|      2 |         | fdsb       |  1334.0
|      3 |      A   | fdsb       |  1334.0
|      4 |  marouan | batata     |    12.0
|      5 |      A   | fdsb       |  1334.0
|      6 |      A   | fdsb       |  1334.0
|      7 |      A   | batbot     |    12.0
+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

```

public ObservableList<AcommandeData> availableAproduitListData() {
    ObservableList<AcommandeData> listData = FXCollections.observableArrayList();
    String sql = "SELECT * FROM credit ";
    try {
        prepare = connect.prepareStatement(sql);
        result = prepare.executeQuery();
        AcommandeData AproduitD;
        while (result.next()) {
            AproduitD = new AcommandeData(result.getInt( columnLabel: "idcredit")
                , result.getString( columnLabel: "username")
                , result.getString( columnLabel: "produitname")
                , result.getDouble( columnLabel: "produitprice")
            );
            listData.add(AproduitD);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return listData;
}

```

La fonction qui nous permet d'afficher notre data bases dans le tableaux

partie User

Button initialiser la page

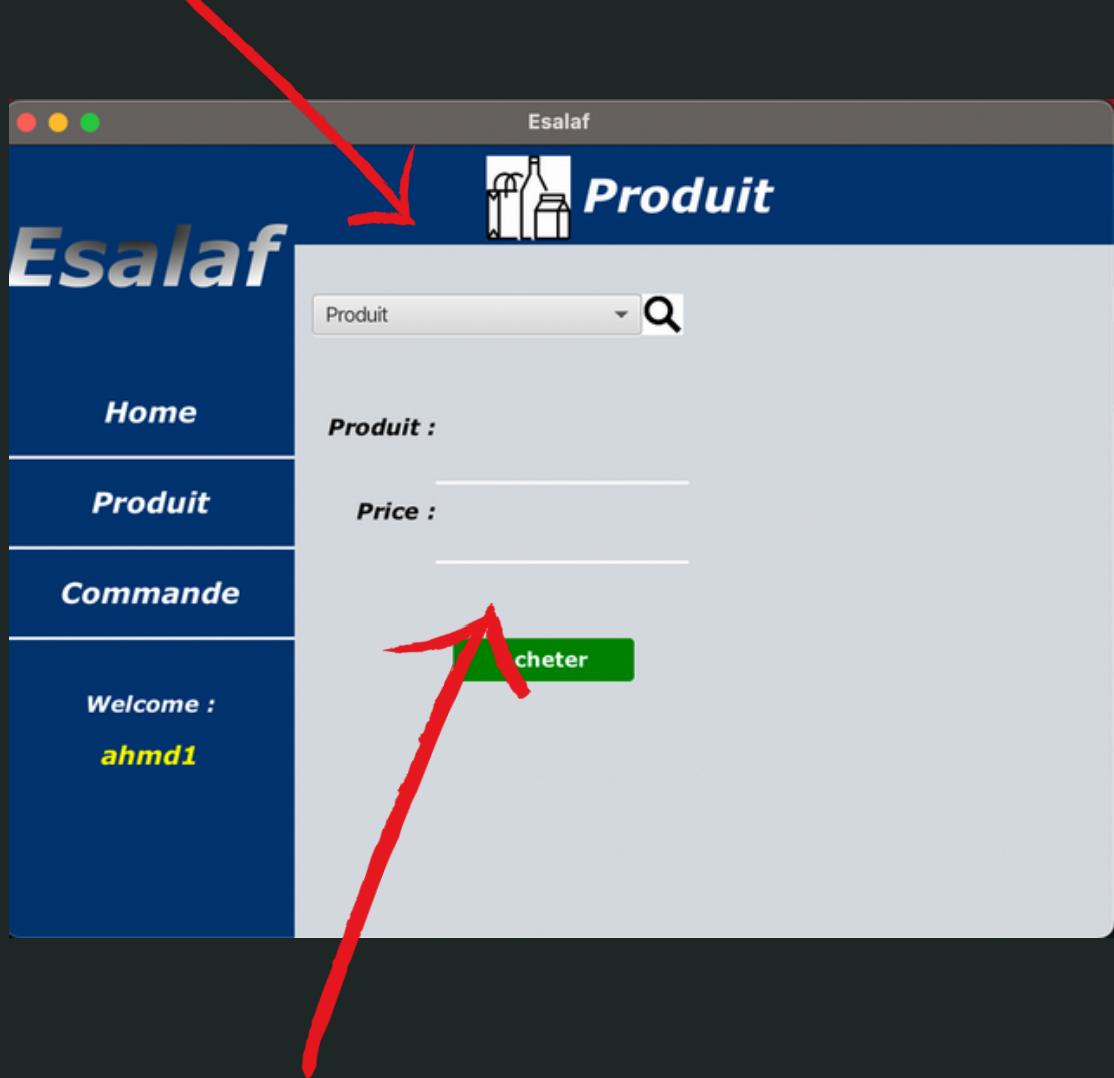
Button pour passer a l'interface produit

Button pour passer a l'interface commande

le nom de utilisateur



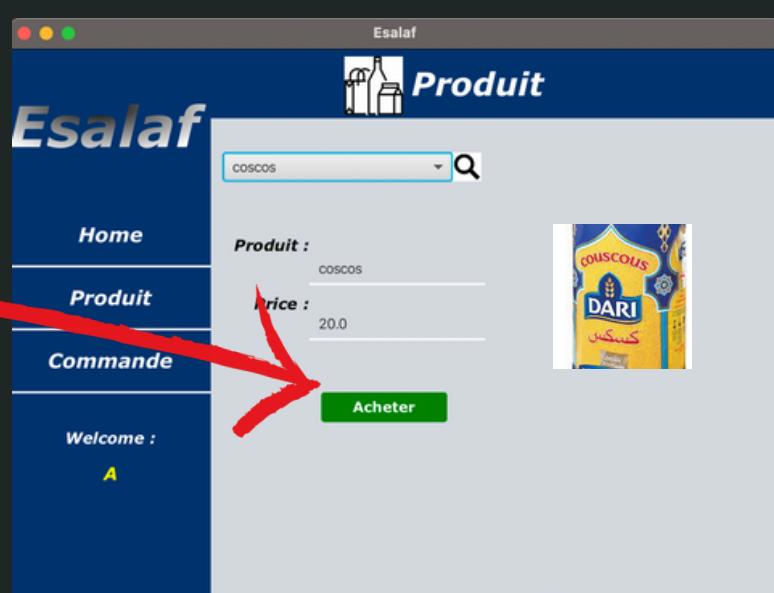
combo pour chercher les produit



Apres choisir produit ,il nous affiche les information sur produit selectionner

Exemple : coscos

Button Acheter



```

1 usage
@FXML
void search_produit() {
    String sql="SELECT produitname ,produitprice, image FROM produit WHERE produitname ="+Combo_searchProduit.getValue()+
    String log ;
    byte byteimg[];
    Blob blob;
    try {
        st =cnx.prepareStatement(sql);
        result=st.executeQuery();
        if (result.next()){
            log=result.getString( columnLabel: "produitname");
            txt_produit.setText(String.valueOf(log));
            txt_price.setText(result.getString( columnLabel: "produitprice"));

            blob=result.getBlob( columnLabel: "image");
            byteimg=blob.getBytes( pos: 1, (int) blob.length());
            Image img = new Image(new ByteArrayInputStream(byteimg));
            imageLog.setImage(img);

        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

```

Fonction qui nous permet de chercher dans combo

```

public void produitAdd() {

    String sql = "INSERT INTO credit (username,produitname, produitprice) VALUES(?, ?, ?) ;";

    try {
        Alert alert;
        if (txt_produit.getText().isEmpty() || txt_price.getText().isEmpty() ) {
            alert = new Alert(Alert.AlertType.ERROR);
            alert.setTitle("Error message");
            alert.setHeaderText(null);
            alert.setContentText("Please fill all blank fields");
            alert.showAndWait();
        } else {

            st = cnx.prepareStatement(sql);
            st.setString( parameterIndex: 1,getData.username);
            st.setString( parameterIndex: 2, txt_produit.getText());
            st.setString( parameterIndex: 3, txt_price.getText());

            st.executeUpdate();

            alert = new Alert(Alert.AlertType.INFORMATION);
            alert.setTitle("information message");
            alert.setHeaderText(null);
            alert.setContentText("Successfully Added");
            alert.showAndWait();
        }
    }
}

```

Fonction qui nous permet de d'acheter produit



Interface commande permet a l'utilisateur de voir leur command qui a deja l'acheter et leur prix (credit)

La fonction qui Affiche a l'utilisateur leur produit qui sont deja acheter (commande)

```

1usage
public ObservableList<AcommandeData> availableAproduitListData() {

    ObservableList<AcommandeData> listData = FXCollections.observableArrayList();

    String sql = "SELECT * FROM credit WHERE username = '" + getData.username + "'";

    try {
        prepare = connect.prepareStatement(sql);
        result = prepare.executeQuery();
        AcommandeData AproduitD;

        while (result.next()) {
            AproduitD = new AcommandeData(result.getInt( columnLabel: "idcredit"),
                result.getString( columnLabel: "username")
                , result.getString( columnLabel: "produitname")
                , result.getDouble( columnLabel: "produitprice"));
            listData.add(AproduitD);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return listData;
}

2 usages
private ObservableList<AcommandeData> availableAproduitList;

```

En conclusion, le projet JavaFX "Esalaf" a été un succès, offrant une solution efficace pour la gestion des prêts dans une institution financière. Grâce à l'utilisation de la technologie JavaFX, une interface utilisateur intuitive a été développée, permettant aux utilisateurs de naviguer facilement dans le système et de réaliser des tâches importantes telles que la création de demandes de prêt, l'approbation de prêts et la gestion des remboursements. De plus, l'architecture du projet a été conçue de manière à offrir une extensibilité future, permettant aux développeurs de facilement ajouter de nouvelles fonctionnalités. Dans l'ensemble, "Esalaf" est un exemple convaincant de la puissance de la technologie JavaFX pour créer des applications desktop modernes et fonctionnelles.

Marouan Daghmoumi
Genie info
Groupe 1
P134160341
20001851

Fin