

Java Mini Projet

Tracking les livreurs



Realiser par : Chahid Anass

Encadre par : Mr El Mokhtar EN-NAIMI

Encadre par : Mr Lotfi Elaachak

TABLE DES MATIERES

INTRODUCTION.....	ERREUR ! SIGNET NON DEFINI.
les outils de travail.....	1
MISE EN PLACE DU PROJET.....	3
CREATION DES INTERFACES.....	4
login.fxml	5
Dashboard.fxml	6
CONNECTION AVEC BD	7
OPERATION SUR LES TABLES	8
CONCLUSION.....	12

Introduction :

Ce rapport décrit le processus impliqué dans la création d'une application desktop de tracking des livreurs.

En utilisant la librairie JDBC, nous avons créé cette application où l'objectif est de saisir ou modifier des données concernant les utilisateurs de l'application, le livreur, les produits, le trajet et les commandes en tant qu'administrateur.

Ce rapport traite de la vue d'ensemble de l'application, y compris la description et l'exécution, puis se concentre sur la conception, décrivant comment les opérations sont mises en œuvre, les fonctions de programmation et les bibliothèques utilisées dans la conception.

Outils de travail :

JAVA :

Java est un langage de programmation orienté objet de haut niveau, basé sur des classes, conçu pour avoir le moins de dépendances d'implémentation possible. Il s'agit d'un langage de programmation à usage général destiné à permettre aux programmeurs d'écrire une fois, de s'exécuter n'importe où, ce qui signifie que le code Java compilé peut s'exécuter sur toutes les plates-formes prenant en charge Java sans avoir besoin de recompiler.

JDBC :

Java Database Connectivity est une interface de programmation d'application pour le langage de programmation Java, qui définit comment un client peut accéder à une base de données. Il s'agit d'une technologie d'accès aux données basée sur Java utilisée pour la connectivité des bases de données Java. Il fait partie de la plate-forme Java Standard Edition, d'Oracle Corporation.

IntelliJ :

IntelliJ IDEA est un environnement de développement intégré (IDE) pour la programmation informatique en Java et d'autres langages de programmation. Il est développé par JetBrains et fournit des fonctionnalités avancées telles que l'analyse de code, la refactorisation de code, la gestion de versions, la prise en charge des tests unitaires, la complétion de code et bien plus encore. IntelliJ IDEA est disponible en version gratuite et payante, avec des fonctionnalités supplémentaires dans la version payante.

JavaFX :

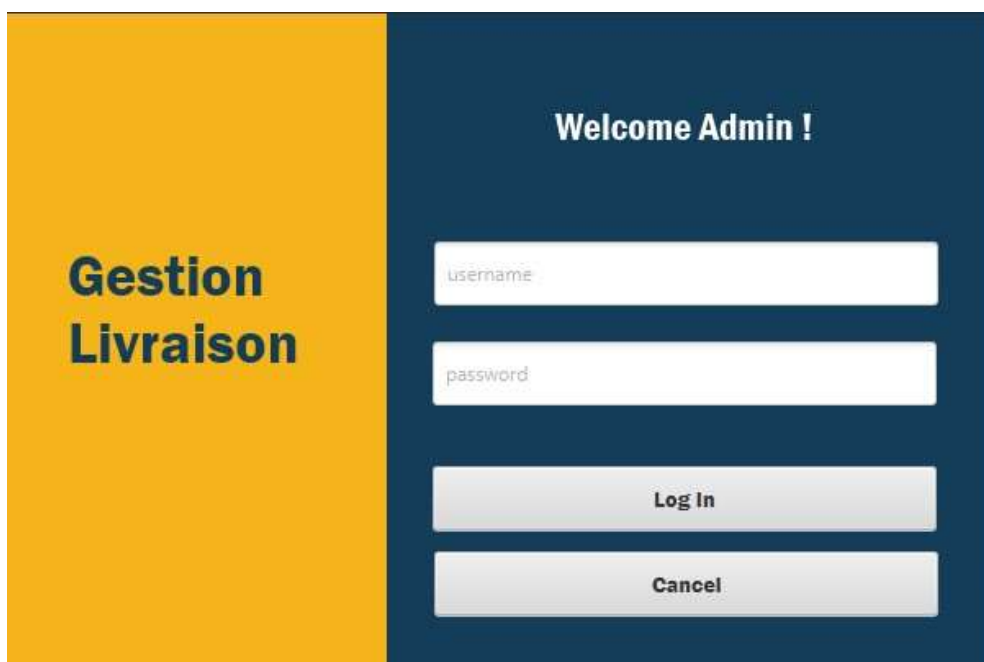
Est une bibliothèque graphique pour Java qui permet de créer des interfaces utilisateur riches et interactives pour les applications de bureau, mobiles et web. Elle offre des composants d'interface utilisateur, des outils de mise en page, des fonctionnalités de gestion des événements, des graphiques 2D et 3D, ainsi que des fonctionnalités multimédias. JavaFX est conçu pour être facile à utiliser, hautement personnalisable et compatible avec les environnements de développement Java existants.

Création de l'interface graphique :

Dans une application desktop le premier chose qui va rencontrer l'utilisateur c'est l'interface,

Dans ce projet là on a utilisé javafx pour créer l'interface de log in, et le dashboard.

Login.fxml :



Pour Contrôler cette interface on crée une class logincontroller.java qui gère l'opération de log in et cancel.

```
public class logcontrol implements Initializable {
    @FXML
    private AnchorPane main_form;

    @FXML
    private TextField id_user;
    @FXML
    private PasswordField id_pass;
    @FXML
    private Button id_log;
    @FXML
    private Button id_close;

    // DATABASE TOOLS
    private Connection connect;
    private PreparedStatement prepare;
    private ResultSet result;

    private double x = 0;
    private double y = 0;
```

```

public void login() {
    String sql = "SELECT * FROM admin WHERE username = ? and password = ?";
    connect = database.connectDb();
    try{
        prepare = connect.prepareStatement(sql);
        prepare.setString(1, id_user.getText());
        prepare.setString(2, id_pass.getText());

        result = prepare.executeQuery();
        Alert alert;

        if(id_user.getText().isEmpty() || id_pass.getText().isEmpty()){
            alert = new Alert(AlertType.ERROR);
            alert.setTitle("Error Message");
            alert.setHeaderText(null);
            alert.setContentText("Please fill all blank fields");
            alert.showAndWait();
        }else{
            if(result.next()){
                alert = new Alert(AlertType.INFORMATION);
                alert.setTitle("Information Message");
                alert.setHeaderText(null);
                alert.setContentText("Successfully Login");
                alert.showAndWait();

                id_log.getScene().getWindow().hide();
                Parent root =
FXMLLoader.load(getClass().getResource("/dashboard.fxml"));
                Stage stage = new Stage();
                Scene scene = new Scene(root);

                root.setOnMousePressed((MouseEvent event) ->{
                    x = event.getSceneX();
                    y = event.getSceneY();
                });

                root.setOnMouseDragged((MouseEvent event) ->{
                    stage.setX(event.getScreenX() - x);
                    stage.setY(event.getScreenY() - y);
                });

                stage.initStyle(StageStyle.TRANSPARENT);
                stage.setScene(scene);
                stage.show();

            }else{
                alert = new Alert(AlertType.ERROR);
                alert.setTitle("Error Message");
                alert.setHeaderText(null);
                alert.setContentText("Wrong Username/Password");
                alert.showAndWait();
            }
        }

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

}

public void close() {
    System.exit(0);
}

@Override
public void initialize(URL url, ResourceBundle rb) {
    // TODO
}

```

}

dashboard.fxml :

id_livreur	Nom	Prenom	Tele
1	chahid	anass	600000000

C'est la fenêtre qui s'affiche lors de la connexion de l'admin. Elle contient un Botton livreur, produit, commande qui représente les tables de notre base de donn  .aussi un botton de log out.

Et sur la m  me interface l'admin peut g  rer toutes les op  rations crud sur les diff  rentes tables.

On peut aussi naviguer entre les tables a cause de ce code :

```
public void switchForm(ActionEvent event) {  
  
    if (event.getSource() == livreur) {  
  
        tab_livr.setVisible(true);  
        tab_pro.setVisible(false);  
        tab_com.setVisible(false);  
  
    } else if (event.getSource() == produit) {  
  
        tab_livr.setVisible(false);  
        tab_pro.setVisible(true);  
        tab_com.setVisible(false);  
  
    } else if (event.getSource() == commande) {  
  
        tab_livr.setVisible(false);  
        tab_pro.setVisible(false);  
        tab_com.setVisible(true);  
  
    }  
}
```

```
}
```

Connection avec la base de données :

```
package com.example.projet_java;
import java.sql.Connection;
import java.sql.DriverManager;
public class database {
    public static Connection connectDb() {

        try{

            Class.forName("com.mysql.cj.jdbc.Driver");

            Connection connect =
DriverManager.getConnection("jdbc:mysql://localhost:3306/delivery", "root", "");
            return connect;
        }catch (Exception e) {e.printStackTrace();}
        return null;
    }
}
```

Opération sur les tables :

On va prendre la table livreur comme exemple :

Welcome back !

Livreur

Produit

commande

Log Out

Made By AC

Gestion Livraison

ID :
Nom :
Prénom :
Téléphone :

Add
Update
Delete

id_livreur	Nom	Prenom	Tele
1	chahid	anass	600000000

Lorsque de l'accès de l'admin a l'interface Dashboard il permet de ajouter ou modifier ou supprimer les élément de la table livreur a l'aide de la class Dashboardcontroll.java

Ajouter :

Welcome back !

Livreur

Produit

commande

Log Out

Made By AC

Gestion Livraison

ID :
Nom :
Prénom :
Téléphone :

Add
Update
Delete

id_livreur	Nom	Prenom	Tele
1	chahid	anass	600000000
2	ahmed	chahid	699887744

Code source :

```
public void addlivreurAdd() {
```



```

Date date = new Date();
java.sql.Date sqlDate = new java.sql.Date(date.getTime());

String sql = "INSERT INTO livreur "
    + "('id_admin','name','prenom','tele)' "
    + "VALUES(?,?,?,?)";

connect = database.connectDb();

try {
    Alert alert;
    if (id_liv_add.getText().isEmpty()
        || nom_liv_add.getText().isEmpty()
        || pre_liv_add.getText().isEmpty()
        || tele_add.getText().isEmpty()
        || getData.path == null || getData.path == "") {
        alert = new Alert(AlertType.ERROR);
        alert.setTitle("Error Message");
        alert.setHeaderText(null);
        alert.setContentText("Successfully Added!");
        alert.showAndWait();
    } else {

        String check = "SELECT id_admin FROM livreur WHERE id_admin = '"
            + id_liv_add.getText() + "'";

        statement = connect.createStatement();
        result = statement.executeQuery(check);

        if (result.next()) {
            alert = new Alert(AlertType.ERROR);
            alert.setTitle("Error Message");
            alert.setHeaderText(null);
            alert.setContentText("livreur ID: " + id_liv_add.getText() + " was
already exist!");
            alert.showAndWait();
        } else {

            prepare = connect.prepareStatement(sql);
            prepare.setString(2, id_liv_add.getText());
            prepare.setString(3, nom_liv_add.getText());
            prepare.setString(4, pre_liv_add.getText());
            prepare.setString(5, tele_add.getText());

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

            prepare.setString(6, uri);
            prepare.setString(7, String.valueOf(sqlDate));
            prepare.executeUpdate();

            String insertInfo = "INSERT INTO livreur "
                + "(id_admin,name,prenom,tele) "
                + "VALUES(?,?,?,?,?,?,?)";

            prepare = connect.prepareStatement(insertInfo);
            prepare.setString(2, id_liv_add.getText());
            prepare.setString(3, nom_liv_add.getText());
            prepare.setString(4, pre_liv_add.getText());
            prepare.setString(5, tele_add.getText());
            prepare.setString(6, "0.0");
            prepare.setString(7, String.valueOf(sqlDate));
            prepare.executeUpdate();

            alert = new Alert(AlertType.INFORMATION);
            alert.setTitle("Information Message");

```

```
        alert.setHeaderText(null);
        alert.setContentText("Successfully Added!");
        alert.showAndWait();

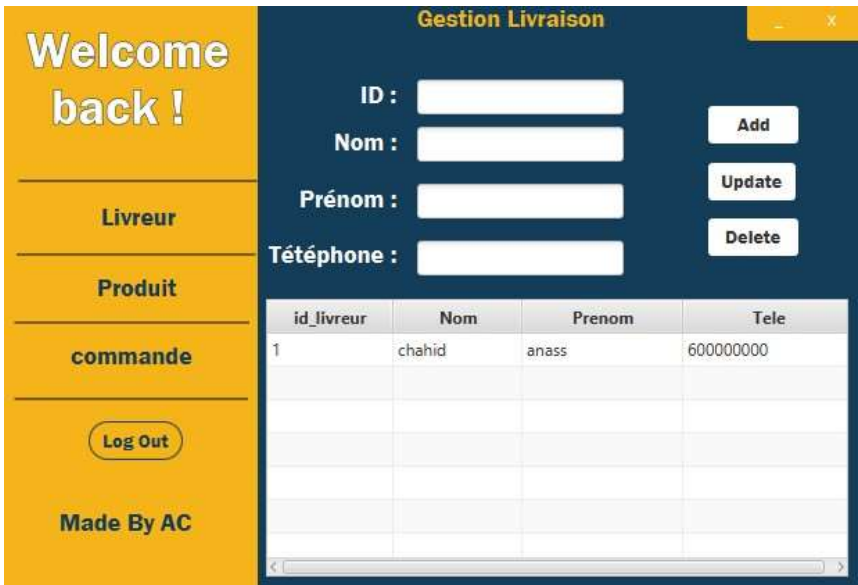
        addlivreurShowListData();

    }

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

}
```

Supprimer :



Opération sur les autres tables :

Table produit :

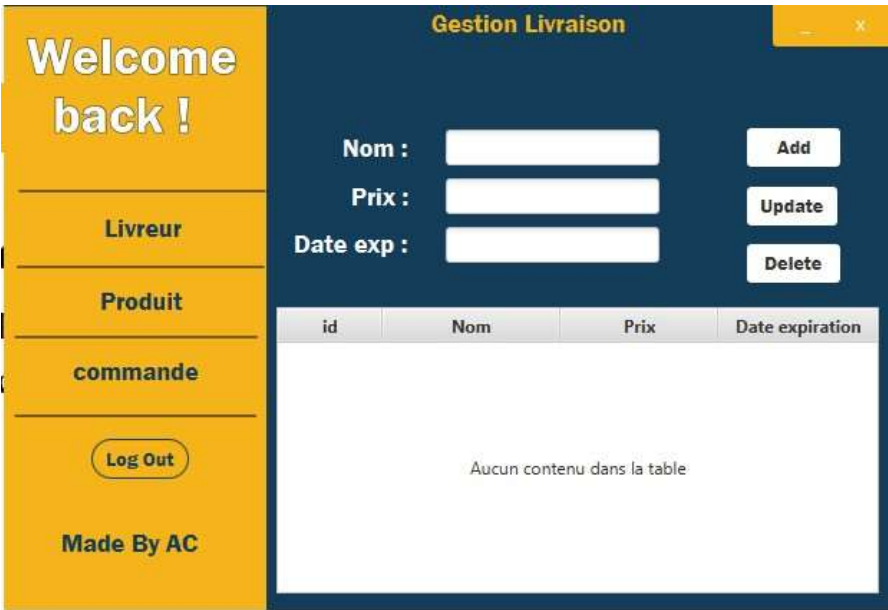


Table commande :

id_com	date	date de livraison	prix total
Aucun contenu dans la table			

Log out de l'application :

Dans la class `dashbordcontroll.java` on trouve le code source qui est responsable a log out de l'admin de son compte.

```
public void logout() {  
  
    Alert alert = new Alert(AlertType.CONFIRMATION);  
    alert.setTitle("Confirmation Message");  
    alert.setHeaderText(null);  
    alert.setContentText("Are you sure you want to logout?");  
    Optional<ButtonType> option = alert.showAndWait();  
    try {  
        if (option.get().equals(ButtonType.OK)) {  
            log_out.getScene().getWindow().hide();  
            Parent root = FXMLLoader.load(getClass().getResource("/login.fxml"));  
            Stage stage = new Stage();  
            Scene scene = new Scene(root);  
  
            root.setOnMousePressed((MouseEvent event) -> {  
                x = event.getSceneX();  
                y = event.getSceneY();  
            });  
            root.setOnMouseDragged((MouseEvent event) -> {  
                stage.setX(event.getScreenX() - x);  
                stage.setY(event.getScreenY() - y);  
                stage.setOpacity(.8);  
            });  
            root.setOnMouseReleased((MouseEvent event) -> {  
                stage.setOpacity(1);  
            });  
  
            stage.initStyle(StageStyle.TRANSPARENT);  
  
            stage.setScene(scene);  
            stage.show();  
        }  
    }  
}
```

On peut aussi fermer l'application ou la minimiser on cliquant sur les bottons x ou –

le premier nous permet de fermer l'app et l'autre de la minimiser :

```
public void close() {  
    System.exit(0);  
}  
public void minimize() {  
    Stage stage = (Stage) main_form.getScene().getWindow();  
    stage.setIconified(true);  
}
```

Conclusion

En guise de conclusion, ce projet nous a permis de mettre en pratique nos connaissances, comme l'implémentation de la programmation orientée objet. Cela nous a également aidé à comprendre le fonctionnement applications desktop,