

POO AVANCE

Projet Java

01/05/2023

—

Mejri Yassmine

—

Bouhleb Dhouha

Interfaces With Scene Builder

User / Admin LOGIN :

The image displays two wireframe mockups of a login interface for a 'Dictionary' application. The top mockup shows a light gray login panel on the left and a blue 'Dictionary' header panel on the right. The bottom mockup shows the blue 'Dictionary' header panel on the left and the light gray login panel on the right.

Top Mockup (Left Panel - Light Gray):

- Icon: Group of three people.
- Form fields:
 - UserName (with user icon)
 - PassWord (with lock icon)
 - Confirm PassWord (with lock icon)
- Buttons:
 - Login As User (blue)
 - [Login As User...](#)

Top Mockup (Right Panel - Blue):

- Icon: Open book.
- Text: Dictionary

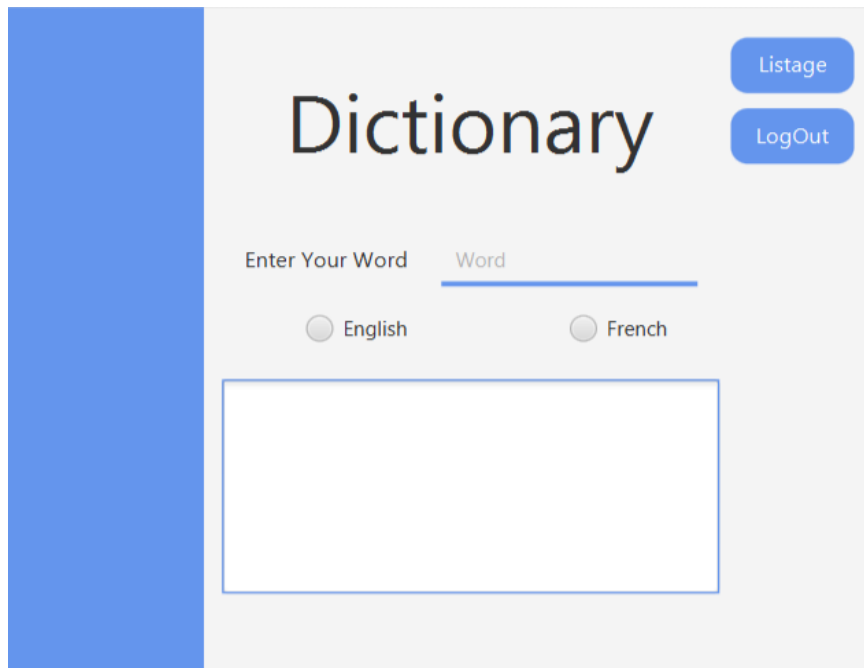
Bottom Mockup (Left Panel - Blue):

- Icon: Open book.
- Text: Dictionary

Bottom Mockup (Right Panel - Light Gray):

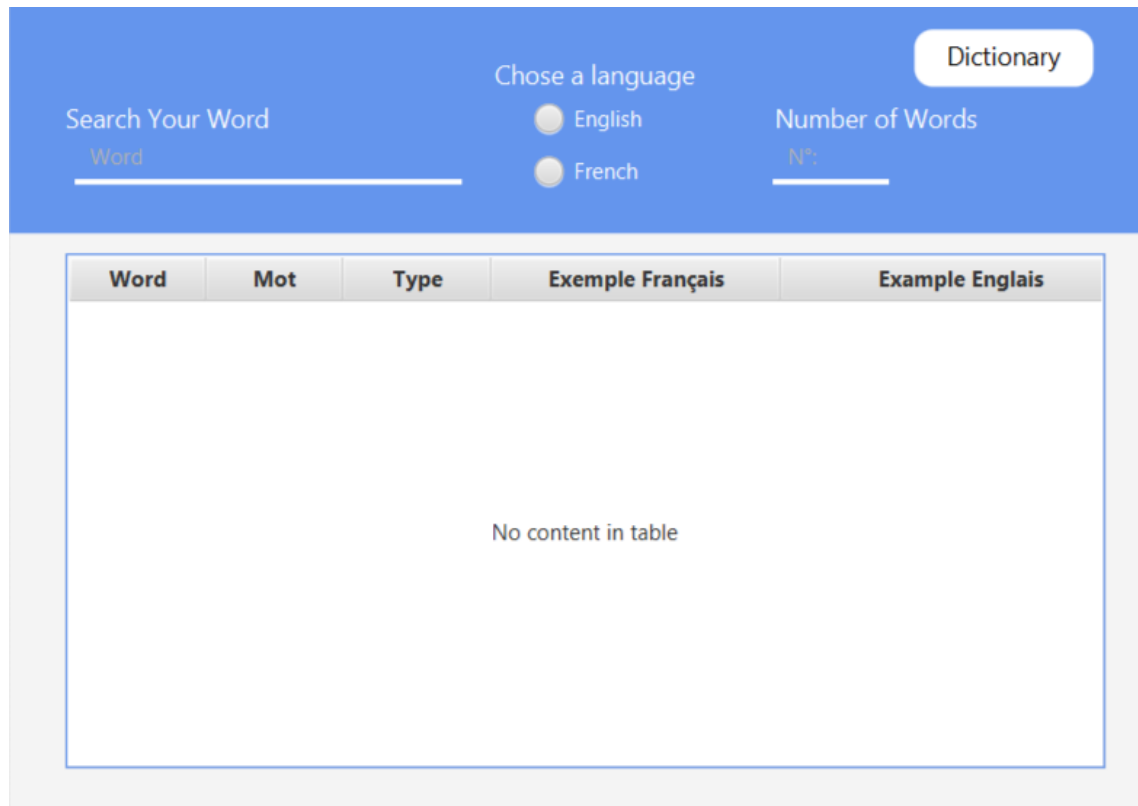
- Icon: Group of three people.
- Form fields:
 - UserName (with user icon)
 - PassWord (with lock icon)
- Buttons:
 - Login As User (blue)
 - Login As Administrator (white)
 - [Create Account For User...](#)

Dictionary Interface



A mockup of a dictionary interface. It features a blue vertical sidebar on the left. The main content area has a light gray background. At the top right, there are two blue buttons labeled 'Listage' and 'LogOut'. The word 'Dictionary' is displayed in a large, dark font. Below it, the text 'Enter Your Word' is followed by a text input field containing the placeholder text 'Word'. Underneath the input field, there are two radio buttons: 'English' (selected) and 'French'. At the bottom, there is a large, empty rectangular box with a blue border, intended for displaying the word's definition.

Listage Interface



A mockup of a listage (list) interface. The top section is a blue header bar. On the left, it says 'Search Your Word' above a text input field with the placeholder 'Word'. In the center, it says 'Chose a language' above two radio buttons: 'English' (selected) and 'French'. On the right, it says 'Number of Words' above a text input field with the placeholder 'N°:'. A white button labeled 'Dictionary' is in the top right corner. Below the header bar is a table with five columns: 'Word', 'Mot', 'Type', 'Exemple Français', and 'Exemple Anglais'. The table body is empty, and the text 'No content in table' is centered within it.

Word	Mot	Type	Exemple Français	Exemple Anglais
No content in table				

Admin Interface

Mot_E

Type_E

Exemple_E

Mot_F

Type_F

Exemple_F

Dictionary

Logout

Mot_E	Type_E	Exemple_E	Synonyme_E
No content in table			

Mot_F	Type_F	Exemple_F	Synonyme_F
No content in table			

Select

Insert

Update

Delete

File

Import/Export Interface

Administrator

Discover the Import and the Export

Word	Mot	Type	Example English	Exemple Français
No content in table				

Import

Export

Code With JavaFx

- Connection to the data base:

```
J App.java      J connectionOracle.java X
dictionary > src > J connectionOracle.java > ...
1  import java.sql.Connection;
2  import java.sql.DriverManager;
3  import java.sql.SQLException;
4
5  public class connectionOracle {
6
7
8      static String user = "SYSTEM";
9      static String password = "0000";
10     static String url = "jdbc:oracle:thin:@localhost:1521:xe";
11     static String driver = "oracle.jdbc.driver.OracleDriver";
12
13
14     public static Connection getCon(){
15         Connection con = null;
16         try {
17             Class.forName(driver);
18             try {
19                 con = DriverManager.getConnection(url,user,password);
20                 System.out.println(x:"Connected with connection oracle"
21             } catch (SQLException e) {
22                 throw new RuntimeException(e);
23             }
24         } catch (ClassNotFoundException e) {
25             throw new RuntimeException(e);
26         }
27         return con;
28     }
29 }
```

- Creation of Tables With Oracle:

```
create table english (mot_e varchar2(50),type_e varchar2(50),exemple_e varchar2(50),synonyme_e varchar2(50));
select * from english;

create table french (mot_f varchar2(50),type_f varchar2(50),exemple_f varchar2(50),synonyme_f varchar2(50));
select * from french;

insert into english values('see','verb','he cant see','voir');
insert into french values('voir','verbe','il ne voit rien','see');
```


- **Classes:**

```
public class French {  
    private String Mot_F;  
    private String Type_F;  
    private String Exemple_F;  
    private String Synonyme_F;  
  
    public String getMot_F() {  
        return Mot_F;  
    }  
    public void setMot_F(String mot_F) {  
        Mot_F = mot_F;  
    }  
    public String getType_F() {  
        return Type_F;  
    }  
    public void setType_F(String type_F) {  
        Type_F = type_F;  
    }  
    public String getExemple_F() {  
        return Exemple_F;  
    }  
    public void setExemple_F(String exemple_F) {  
        Exemple_F = exemple_F;  
    }  
    public String getSynonyme_F() {  
        return Synonyme_F;  
    }  
    public void setSynonyme_F(String synonyme_F) {  
        Synonyme_F = synonyme_F;  
    }  
}
```

```
public class English {  
    private String Mot_E;  
    private String Type_E;  
    private String Exemple_E;  
    private String Synonyme_E;  
  
    public String getMot_E() {  
        return Mot_E;  
    }  
    public void setMot_E(String mot_E) {  
        Mot_E = mot_E;  
    }  
    public String getType_E() {  
        return Type_E;  
    }  
    public void setType_E(String type_E) {  
        Type_E = type_E;  
    }  
    public String getExemple_E() {  
        return Exemple_E;  
    }  
    public void setExemple_E(String exemple_E) {  
        Exemple_E = exemple_E;  
    }  
    public String getSynonyme_E() {  
        return Synonyme_E;  
    }  
    public void setSynonyme_E(String synonyme_E) {  
        Synonyme_E = synonyme_E;  
    }  
}
```

- **Translation:**

```
@FXML  
void btnEc(ActionEvent event) {  
    String query = "Select * from English,French where Synonyme_E='"+word.getText()+"';"  
    con = connectionOracle.getCon();  
    try{  
        stmt = con.prepareStatement(query);  
        rs= stmt.executeQuery();  
        while(rs.next()){  
            disc.setText(" Mot: "+rs.getString(columnLabel:"Mot_E")+"\n Type : "+rs.getString(columnLabel:"type_E")+"\n Exemple Anglais : "  
            +rs.getString(columnLabel:"Exemple_E")+"\n Exemple Français : "+rs.getString(columnLabel:"Exemple_F"));  
        }  
        disc.setEditable(false);  
        stmt.executeUpdate();  
    }catch(SQLException e){  
        throw new RuntimeException(e);  
    }  
}  
  
@FXML  
void btnFc(ActionEvent event) {  
    String query = "Select * from English,French where Synonyme_F='"+word.getText()+"';"  
    con = connectionOracle.getCon();  
    try{  
        stmt = con.prepareStatement(query);  
        rs= stmt.executeQuery();  
        while(rs.next()){  
            disc.setText(" Word: "+rs.getString(columnLabel:"Mot_F")+"\n Type : "+rs.getString(columnLabel:"type_F")+"\n French Example : "  
            +rs.getString(columnLabel:"Exemple_F")+"\n English Example : "+rs.getString(columnLabel:"Exemple_E"));  
        }  
        disc.setEditable(false);  
        stmt.executeUpdate();  
    }catch(SQLException e){  
        throw new RuntimeException(e);  
    }  
}
```

- Getting all the translation of both languages:

```
public ObservableList<English> getEnglish(){
    ObservableList<English> english = FXCollections.observableArrayList();

    String query = "Select * from English";
    con = connectionOracle.getCon();
    try{
        stmt = con.prepareStatement(query);
        rs= stmt.executeQuery();
        while(rs.next()){
            English en = new English();
            en.setMot_E(rs.getString(columnLabel:"Mot_E"));
            en.setType_E(rs.getString(columnLabel:"Type_E"));
            en.setExemple_E(rs.getString(columnLabel:"Exemple_E"));
            en.setSynonyme_E(rs.getString(columnLabel:"Synonyme_E"));
            english.add(en);
        }
    }catch(SQLException e){
        throw new RuntimeException(e);
    }
    return english;
}

public void showEnglish(){
    ObservableList<English> list = getEnglish();
    tbl1.setItems(list);
    col_1.setCellValueFactory(new PropertyValueFactory<English, String>("Mot_E"));
    col_2.setCellValueFactory(new PropertyValueFactory<English, String>("Type_E"));
    col_3.setCellValueFactory(new PropertyValueFactory<English, String>("Exemple_E"));
    col_4.setCellValueFactory(new PropertyValueFactory<English, String>("Synonyme_E"));
}
```

```
public ObservableList<French> getFrench(){
    ObservableList<French> french = FXCollections.observableArrayList();
    String query = "Select * from French";
    con = connectionOracle.getCon();
    try{
        stmt = con.prepareStatement(query);
        rs= stmt.executeQuery();
        while(rs.next()){
            French fr = new French();
            fr.setMot_F(rs.getString(columnLabel:"Mot_F"));
            fr.setType_F(rs.getString(columnLabel:"Type_F"));
            fr.setExemple_F(rs.getString(columnLabel:"Exemple_F"));
            fr.setSynonyme_F(rs.getString(columnLabel:"Synonyme_F"));
            french.add(fr);
        }
    }catch(SQLException e){
        throw new RuntimeException(e);
    }
    return french;
}

public void showFrench(){
    ObservableList<French> list = getFrench();
    tbl2.setItems(list);
    col_11.setCellValueFactory(new PropertyValueFactory<French, String>("Mot_F"));
    col_22.setCellValueFactory(new PropertyValueFactory<French, String>("Type_F"));
    col_33.setCellValueFactory(new PropertyValueFactory<French, String>("Exemple_F"));
    col_44.setCellValueFactory(new PropertyValueFactory<French, String>("Synonyme_F"));
}
```

- Setting New translation of both languages:

```
@FXML
void btn2c(ActionEvent event) {
    String insertE = "insert into English(Mot_E,Type_E,Exemple_E,Synonyme_E) values(?,?,?,?)";
    String insertF = "insert into French(Mot_F,Type_F,Exemple_F,Synonyme_F) values(?,?,?,?)";
    con = connectionOracle.getCon();

    try{
        stmt = con.prepareStatement(insertE);
        stmt.setString(parameterIndex:1, txt1.getText());
        stmt.setString(parameterIndex:2, txt2.getText());
        stmt.setString(parameterIndex:3, txt3.getText());
        stmt.setString(parameterIndex:4, txt4.getText());
        stmt.executeUpdate();

        stmt1 = con.prepareStatement(insertF);
        stmt1.setString(parameterIndex:1, txt4.getText());
        stmt1.setString(parameterIndex:2, txt5.getText());
        stmt1.setString(parameterIndex:3, txt6.getText());
        stmt1.setString(parameterIndex:4, txt1.getText());
        stmt1.executeUpdate();

        showEnglish();
        showFrench();
        clear();
    }catch (SQLException e){
        throw new RuntimeException(e);
    }
}
```

- Updating translations:

```
@FXML
void btn3c(ActionEvent event) {
    String updateE="update English set mot_E=?, type_E=?, exemple_E=?, synonyme_E=? where mot_E=?";
    String updateF="update French set mot_F=?, type_F=?, exemple_F=?, synonyme_F=? where mot_F=?";
    con=connectionOracle.getCon();
    try{
        stmt=con.prepareStatement(updateE);
        stmt.setString(parameterIndex:1, txt1.getText());
        stmt.setString(parameterIndex:2, txt2.getText());
        stmt.setString(parameterIndex:3, txt3.getText());
        stmt.setString(parameterIndex:4, txt4.getText());

        stmt.setString(parameterIndex:5,txt1.getText() );
        stmt.executeUpdate();
        stmt1 = con.prepareStatement(updateF);
        stmt1.setString(parameterIndex:1, txt4.getText());
        stmt1.setString(parameterIndex:2, txt5.getText());
        stmt1.setString(parameterIndex:3, txt6.getText());
        stmt1.setString(parameterIndex:4, txt1.getText());

        stmt1.setString(parameterIndex:5, txt4.getText());
        stmt1.executeUpdate();
        showEnglish();
        showFrench();
    }catch(SQLException e){
        throw new RuntimeException();
    }
}
```

- Deleting translations:

```
@FXML
void btn4c(ActionEvent event) {
    String deleteE = "delete from English where Mot_E=?";
    String deleteF = "delete from French where Mot_F=?";
    con = connectionOracle.getCon();
    try{
        stmt = con.prepareStatement(deleteE);
        stmt.setString(parameterIndex:1,txt1.getText());
        stmt.executeQuery();

        stmt1 = con.prepareStatement(deleteF);
        stmt1.setString(parameterIndex:1,txt4.getText());
        stmt1.executeQuery();
        showFrench();
        showEnglish();
        clear();
    }catch(SQLException e){
        throw new RuntimeException(e);
    }
}

void clear(){
    txt1.setText(null);
    txt2.setText(null);
    txt3.setText(null);
    txt4.setText(null);
    txt5.setText(null);
    txt6.setText(null);
    btn2.setDisable(false);
}
```

- Getting Data:

```
@FXML
void getDataE(MouseEvent event) {
    English en =tbl1.getSelectionModel().getSelectedItem();

    txt1.setText(en.getMot_E());
    txt2.setText(en.getType_E());
    txt3.setText(en.getExemple_E());

    btn2.setDisable(true);
}

@FXML
void getDataF(MouseEvent event) {
    French fr =tbl2.getSelectionModel().getSelectedItem();
    txt4.setText(fr.getMot_F());
    txt5.setText(fr.getType_F());
    txt6.setText(fr.getExemple_F());

    btn2.setDisable(true);
}
```

- **Class Objet:**

```
J App.java  J Objet.java X
dictionary > src > J Objet.java > Example_F
1  import java.io.Serializable;
2
3  public class Objet implements Serializable {
4      private String Mot_E;
5      private String Mot_F;
6      private String Type_E;
7      private String Exemple_E;
8      private String Exemple_F;
9      public Objet() {
10     }
11
12     public Objet(String mot_E, String mot_F, String type_E, String exemple_E, String exemple_F) {
13         Mot_E = mot_E;
14         Mot_F = mot_F;
15         Type_E = type_E;
16         Exemple_E = exemple_E;
17         Exemple_F = exemple_F;
18     }
19
20     public String getMot_E() {
21         return Mot_E;
22     }
23     public void setMot_E(String mot_E) {
24         Mot_E = mot_E;
25     }
26
27     public String getMot_F() {
28         return Mot_F;
29     }
30     public void setMot_F(String mot_F) {
31         Mot_F = mot_F;
32     }
33
34     public String getType_E() {
35         return Type_E;
36     }
37     public void setType_E(String type_E) {
```

- **Getting object from data and write it in a list:**

```
public ObservableList<Objet> getObject(){
    ObservableList<Objet> object = FXCollections.observableArrayList();
    String query = "select * from French,English where mot_E=synonyme_F and mot_F=synonyme_E";
    con = connectionOracle.getConnection();
    try{
        stmt = con.prepareStatement(query);
        rs= stmt.executeQuery();
        while(rs.next()){
            Objet ob = new Objet(rs.getString(columnLabel:"Mot_E"), rs.getString(columnLabel:"Mot_F"), rs.getString(columnLabel:"Type_F"),
            rs.getString(columnLabel:"Exemple_E"),rs.getString(columnLabel:"Exemple_F"));
            object.add(ob);
        }
    }catch(SQLException e){
        throw new RuntimeException(e);
    }
    return object;
}

public void Affichage(){
    ObservableList<Objet> list = getObject();
    aff.setItems(list);
    word.setCellValueFactory(new PropertyValueFactory<Objet, String>("Mot_E"));
    mot.setCellValueFactory(new PropertyValueFactory<Objet, String>("Mot_F"));
    type.setCellValueFactory(new PropertyValueFactory<Objet, String>("Type_E"));
    expE.setCellValueFactory(new PropertyValueFactory<Objet, String>("Exemple_E"));
    expF.setCellValueFactory(new PropertyValueFactory<Objet, String>("Exemple_F"));
}
```

- Export and import data:

```
@FXML
void exportbtn(ActionEvent event) throws Exception{
    String ch="";
    Object obj = null;
    try{
        File file =new File(pathname+"import1.txt");
        ObjectInputStream ObjectInputStream = new ObjectInputStream(new FileInputStream(file));
        while ((obj=ObjectInputStream.readObject())!=null){
            if (obj instanceof Object){
                ch=ch+((Object) obj).Affichage();
            }
            expT.setText(ch);
        }
        ObjectInputStream.close();
    }catch (Exception ex) {
        System.out.println(ex.getMessage());
    }
}

@FXML
void importbtn(ActionEvent event) throws Exception {
    try{
        File file =new File(pathname+"import1.txt");
        Affichage();
        ObservableList<Object> list = getObject();
        ObjectOutputStream ObjectOutputStream = new ObjectOutputStream(new FileOutputStream(file));
        for (int i =0; i<list.size();i++){
            ObjectOutputStream.writeObject(list.get(i));
            System.out.println(list.get(i));
        }
        ObjectOutputStream.close();
    }catch (Exception ex){
        throw new RuntimeException(ex);
    }
}
```

- Creating accounts (Users):

```
@FXML
void loginbtnc(ActionEvent event) {
    String insert = "insert into diclogin(username,password) values(?,?)";
    con = connectionOracle.getCon();
    try{
        String pas=passtxtcr.getText().toString();
        String conf=passconftxtcr.getText().toString();

        if((pas.compareTo(conf))==0){
            stmt = con.prepareStatement(insert);
            stmt.setString(parameterIndex:1, usertxtcr.getText());
            stmt.setString(parameterIndex:2, passtxtcr.getText());
            stmt.executeUpdate();

            FXMLLoader loader = new FXMLLoader(getClass().getResource(name:"loginpage.fxml"));
            loader.getController();
            Parent root = loader.load();
            Stage window= (Stage) linklogin.getScene().getWindow();
            Scene scene = new Scene(root);
            window.setScene(scene);
            window.show();
        }else{
            Alert alert1 = new Alert(Alert.AlertType.ERROR);
            alert1.setHeaderText(headerText:" Invalid Password!");
            alert1.showAndWait();
            usertxtcr.clear();
            passtxtcr.clear();
            passconftxtcr.clear();
        }
    }catch (Exception e) {
        e.printStackTrace();
    }
}
```

- Login as user:

```
@FXML
void loginbtnuc(ActionEvent event) {

    String query = "Select * from login where username='"+usertxt.getText()+"' and password= '"+passtxt.getText()+"'";
    con = connectionOracle.getCon();
    try{
        stmt = con.prepareStatement(query);
        rs= stmt.executeQuery();
        while(rs.next()){
            if((usertxt.getText().compareTo(rs.getString(columnLabel:"username"))==0)
            && (passtxt.getText().compareTo(rs.getString(columnLabel:"password"))==0) ){
                FXMLLoader loader = new FXMLLoader(getClass().getResource("user.fxml"));
                loader.getController();
                Parent root = loader.load();
                Stage window= (Stage) loginbtncu.getScene().getWindow();
                Scene scene = new Scene(root);
                window.setScene(scene);
                window.show();
            }else{
                Alert alert2 = new Alert(Alert.AlertType.ERROR);
                alert2.setHeaderText(" Invalid Username and Password!");
                alert2.showAndWait();
                usertxt.clear();
                passtxt.clear();
            }
        }
    }catch(Exception e){
        throw new RuntimeException(e);
    }
}
```

- We used two types of files (object and text):

