

BANK

Start

Exit

```
package bankusersgui;

import java.io.IOException;
import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

public class BankUsersGUI extends Application {

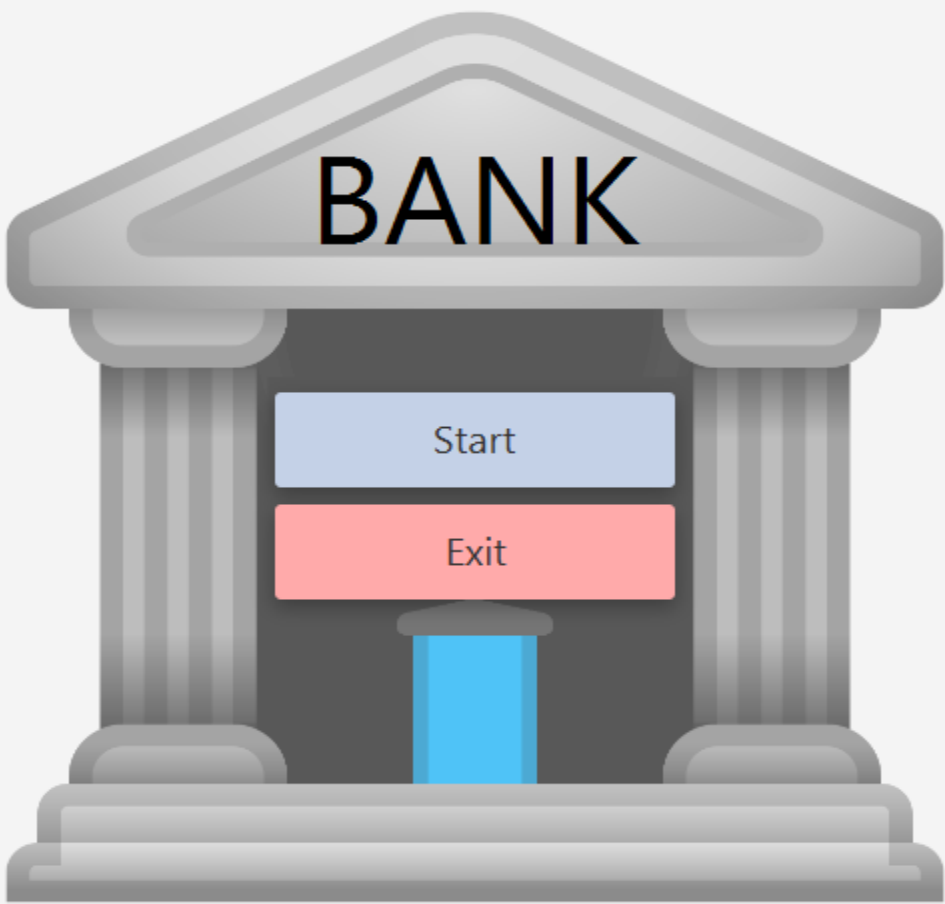
    @Override
    public void start(Stage stage) throws Exception {
        Parent root = FXMLLoader.load(getClass().getResource("start.fxml"));

        Scene scene = new Scene(root);

        stage.setScene(scene);
        stage.show();
    }

    public static void main(String[] args) throws IOException {
        launch(args);
    }
}
```

كلاس BankUsersGUI هو الكلاس الأساسي فيه الدالة start وفيها يتم عرض ال scene الأولى الموضحة بالصورة



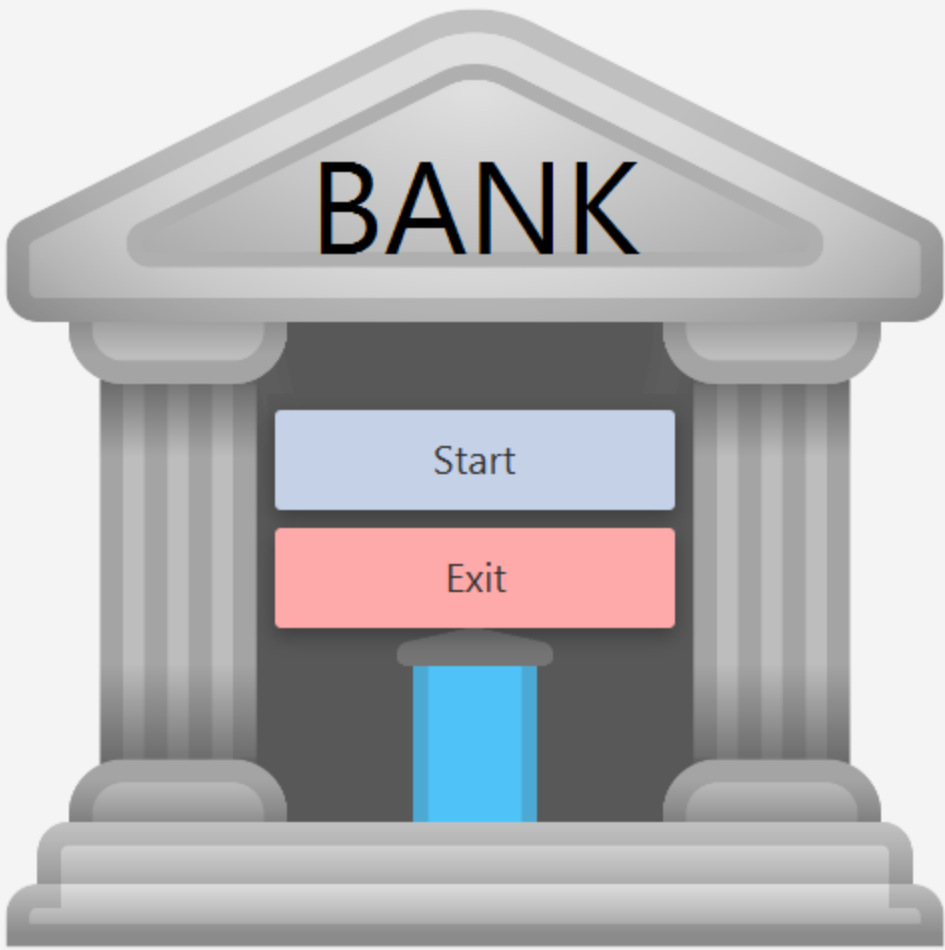
```
public class StartController implements Initializable {

    @FXML
    private JFXButton startB;
    @FXML
    private AnchorPane main;
    @FXML
    private JFXButton exitB;
    @FXML
    private Label bank;
    @FXML
    private JFXButton test;
    File file = new File("remember.txt");
    mohamadBuffered BFR = new mohamadBuffered(file);
    DB DB = new DB();
    Scens scene=new Scens();
    @Override
    public void initialize(URL url, ResourceBundle rb) {
        exitB.setOnAction((ActionEvent event) -> {
            System.exit(0);
        });
        main.setStyle("-fx-background-image: url('file:C:/Users/srdam/Documents/NetBeansProjects/BankUsersGUI/bank.png');-fx-b
        if (SettingController.lang) {
            bank.setText("المصرف");
            bank.setLayoutX(160);
            bank.setFont(new Font(50));
            startB.setText("ابدء");
            exitB.setText("خروج");
        }
    }

}
```

Activate Windows

الكلاس StartController هو الكلاس المرتبط بال scene الأولى وفيه نربط أولا العناصر ثم المتغير file يشير إلى الملف remember والذي يحوي رقم بطاقة وكلمة مرور آخر مستخدم قام بتسجيل الدخول وقام بتحديد خيار rememberme وتعريف بعض المتغيرات من كلاسات أخرى بعد ذلك دالة initialize وفيها event للزر exitB يقوم فيه بإغلاق البرنامج عند الضغط عليه وإضافة style إلى ال main وهو مدير التنسيق وهذا ال style هو الصورة ثم التأكد إذا كان المتغير lang من الكلاس SettingController هو true هذا يعني أن لغة البرنامج هي العربية فيقوم بتغيير النص



@FXML

```
private void start(ActionEvent event) throws SQLException, FileNotFoundException, IOException {  
    if (file.exists()) {  
        String cardnumber = BFR.readline(1);  
        String pass = BFR.readline(2);  
        if (DB.signin(cardnumber, pass).getRow() == 1) {  
            SigninController.cardnumber = Integer.parseInt(cardnumber);  
            SettingController.s = DB.getcolor(Integer.parseInt(cardnumber));  
            SettingController.lang = DB.getlang(SigninController.cardnumber);  
            scene.changescene("/chois/chois.fxml", event);  
        } else {  
            scene.changescene("/signin/signin.fxml", event);  
        }  
    } else {  
        scene.changescene("/signin/signin.fxml", event);  
    }  
}
```

وهنا نتابع في كلاس StartController الدالة start والتي تنفذ عند الضغط على زر start فيتأكد إذا كان الfile الذي يشير إلى الملف remember موجود ويخزن أول سطرين منه في المتغيرين cardnumber و pass ويتأكد إذا كان هذان المتغيران يمثلان رقم بطاقة وكلمة سر لأحد المستخدمين في الdatabase فإذا كانا صحيحين يخزن الcardnumber في المتغير cardnumber من الكلاس SigninController والذي يمثل رقم بطاقة المستخدم الحالي ويخزن القيمة الموجودة في العمود color من الdatabase في المتغير s من الكلاس SettingController والقيمة الموجودة في العمود lang من الdatabase في المتغير lang من الكلاس SettingController ومن ثم يستدعي الدالة changescene من الكلاس Scenes والتي تغير بين الواجهات ليغير إلى الواجهة الرئيسية للمستخدم وإذا كان الملف غير موجود أو البيانات الموجودة فيه خاطئة يقوم بنقله إلى الواجهة signin والتي هي واجهة تسجيل الدخول

```

public class mohamadBuffered {

    private BufferedWriter BFW = null;
    private BufferedReader BFR = null;
    private File f = null;
    private List<String> alllines = null;
    private Boolean compleat = false;

    public mohamadBuffered(File f) {
        this.f = f;
    }

    public mohamadBuffered(File f, Boolean b) {
        compleat = b;
        this.f = f;
    }

    public List<String> readalllines() throws IOException {
        String line;
        BFR = new BufferedReader(new FileReader(f));
        alllines = new ArrayList();
        while (true) {
            line = BFR.readLine();
            if (line == null) {
                break;
            }
            alllines.add(line);
        }
        return alllines;
    }
}

```

```

public String readline(int indexoffline) throws IOException {
    String line = "";
    try {
        line = readalllines().get(indexoffline - 1);
    } catch (Exception e) {
        System.out.println("there are no line with this index");
    }
    return line;
}

public void replace(int index, String newvalue) throws IOException {
    try {
        alllines = readalllines();
        alllines.remove(index);
        alllines.add(index, newvalue);
        BFW = new BufferedWriter(new FileWriter(f));
        for (String line : alllines) {
            BFW.write(line + "\n");
        }
    } catch (Exception e) {
        System.out.println(e.getMessage());
    } finally {
        if (BFW != null) {
            BFW.close();
        }
    }
}

public void write(String... a) throws IOException {
    BFW = new BufferedWriter(new FileWriter(f, compleat));
    for (String b : a) {
        BFW.write(b + "\n");
    }
    BFW.close();
}
}

```

هنا الكلاس mohamadBuffered وهو كلاس للقراءة ولكتابة والتعديل على الملف فيه كونستكران الأول يأخذ باراميتر واحد نوعه file والثاني يأخذ باراميتران الأول نوعه file والثاني نوعه boolean وفيه الدوال الآتية:

١-readalllines والتي تقوم بقراءة جميع أسطر الملف وترجع List فيها هذه الأسطر على شكل String

٢-readline وهي تأخذ باراميتر واحد نوعه int وهو يحدد رقم السطر المراد قراءته وترجع String يمثل بيانات هذا السطر

٣-replace وهي دالة لتبديل بيانات في الملف والتي تأخذ باراميتران الأول int يمثل السطر المراد تبديله والثاني String يمثل القيمة الجديدة

٤-write وهي تأخذ عددا غير محدد من الString كباراميترات وتقوم بكتابتها في الملف كل في سطر

```

public class Scens {

    public void scene(String a, Button b) {
        b.setOnAction((e) -> {
            try {
                changescene(a, e);
            } catch (IOException ex) {
                System.out.println(ex.getMessage());
            }
        });
    }

    public void changescene(String a,(ActionEvent e) throws IOException {
        Parent Parent;
        Parent = FXMLLoader.load(getClass().getResource(a));

        Scene scene = new Scene(Parent, 500, 500);

        Stage Stage = (Stage) ((Node) (e.getSource())).getScene().getWindow();

        Stage.setScene(scene);
    }

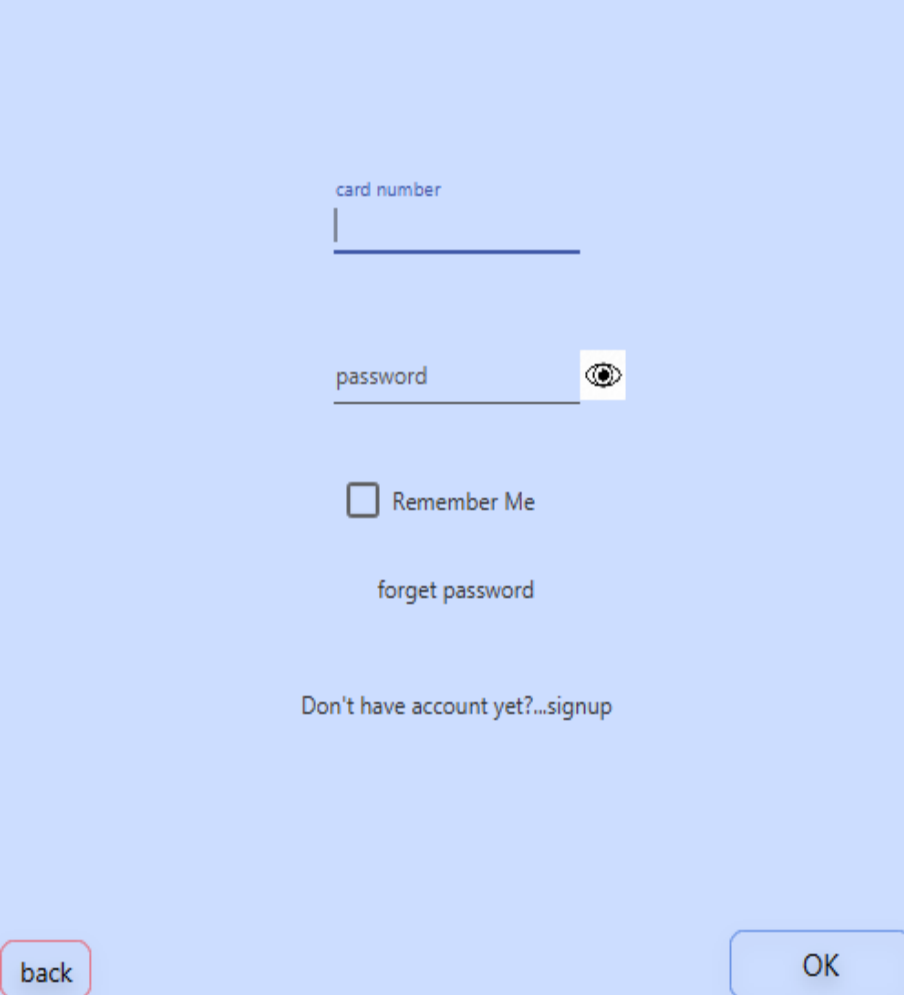
    public Integer chekpass(String a) {
        int b = 0;
        final String small = "abcdefghijklmnopqrstuvwxyz";
        final String big = small.toUpperCase();
        final String num = "0123456789";
        final String SC = "!#$%&'()*+,-./:;<=>?@[\\]^_`{|}~";
        for (Character Char: small.toCharArray()) {
            if (a.contains(Char.toString())) {
                b++;
                break;}}
        for (Character Char: big.toCharArray()) {
            if (a.contains(Char.toString())) {
                b++;
                break;}}
        for (Character Char: num.toCharArray()) {
            if (a.contains(Char.toString())) {
                b++;
                break;}}
        for (Character Char: SC.toCharArray()) {
            if (a.contains(Char.toString())) {
                b++;
                break;}}

        if (a.length()
            < 8) {
            b = 0;} else if (a.length() >= 8) {
            b++;
        }
        return b;
    }
}

```

الكلاس Scens فيه الدالة changescene والتي تأخذ في باراميتراتها متغيرين الأول String والثاني(ActionEvent) وهي تقوم بتغيير الواجهة إلى ملف ال fxml المشار إليه بالمتغير a والدالة scene التي تأخذ في باراميتراتها متغير a نوعه String ومتغير b نوعه button وهي تقوم باستدعاء الدالة changescene عند النقر على الزر b

الدالة chekpass تأخذ في باراميتراتها متغير a نوعه String والتي تقوم بالتأكد من شروط كلمة المرور والتي هي أن يحوي حرف صغير وحرف كبير ورقم وحرف من الحروف الخاصة ويكون طولها أكثر من ثمانية أحرف وترجع integer يحمل عدد النقاط التي حصلت عليها كلمة السر والتي يجب أن تكون ٥ ليتم قبولها



```
public class SigninController implements Initializable {

    @FXML
    private AnchorPane pane;
    @FXML
    private JFXTextField cardT;
    @FXML
    private JFXTextField showpassT;
    @FXML
    private JFXPasswordField passT;
    @FXML
    private JFXButton ok;
    @FXML
    private JFXButton back;
    @FXML
    private JFXButton showB;
    @FXML
    private Button forget;
    @FXML
    private Button signup;
    @FXML
    private Label incorrect;
    @FXML
    private JFXCheckBox Remember;

    signinmethods remember = new signinmethods();
    Scens scene = new Scens();
    public static Integer cardnumber = 0;
    DB DB = new DB();
```

```
@Override
public void initialize(URL url, ResourceBundle rb) {
    pane.setStyle(s);
    scene.scene("/bankusersgui/start.fxml", back);
    scene.scene("/forget/forget.fxml", forget);
    scene.scene("/signup/signup.fxml", signup);
    forget.setBackground(Background.EMPTY);
    signup.setBackground(Background.EMPTY);
    showB.setStyle("-fx-background-image: url('file:C:/Users/srdam/Documents/NetBeansProjects/BankUsersGUI/show.png');-fx-
if (SettingController.lang) {
    cardT.setPromptText("رقم البطاقة");
    passT.setPromptText("كلمة السر");
    ok.setText("انطلق");
    back.setText("رجوع");
    forget.setText("نسيت كلمة المرور");
    forget.setLayoutX((400 - forget.getPrefWidth()) / 2);
    signup.setText("إنشاء حساب جديد");
    signup.setLayoutX((400 - signup.getPrefWidth()) / 2);
    incorrect.setText("خطأ في رقم البطاقة أو كلمة المرور");
    Remember.setText("تذكرني");
}
}
```

الواجهة الثانية هي واجهة تسجيل الدخول المرتبطة بالكلاس SigninController الكلاس SigninController فيه أولا ربط عناصر الواجهة بمتغيرات في الكلاس ثم تعريف متغيرات من كلاسات أخرى وتعريف المتغير cardnumber الذي نوعه integer والذي يحمل رقم البطاقة للمستخدم الحالي ثم دالة initialize وفيها أولا تعديل الـ style لمدير التنسيق وكود الـ css هو من المتغير s من الكلاس SettingController و ثم استدعاء دوال scene من الكلاس Scens للتبديل بين الواجهات بحسب الزر المضغوط ثم إضافة صورة إلى الزر showB وتغيير النص بحسب اللغة

card number

password

☐ Remember Me

forget password

Don't have account yet?...signup

back

OK

```

@FXML
private void chanebutton(MouseEvent event) {
    if (event.getSource().equals(signup)) {
        signup.setStyle(signup.getStyle() + "-fx-text-fill:#0000ff;-fx-underline:true;");
    } else if (event.getSource().equals(forget)) {
        forget.setStyle(forget.getStyle() + "-fx-text-fill:#0000ff;-fx-underline:true;");
    }
}

@FXML
private void retornbutton(MouseEvent event) {
    if (event.getSource().equals(signup)) {
        signup.setStyle(signup.getStyle() + "-fx-text-fill:black;-fx-underline:false;");
    } else if (event.getSource().equals(forget)) {
        forget.setStyle(forget.getStyle() + "-fx-text-fill:black;-fx-underline:false;");
    }
}

@FXML
private void show(ActionEvent e) {
    if (passT.isVisible()) {
        showpassT.setText(passT.getText());
        passT.setVisible(false);
        showpassT.setVisible(true);
        showB.setStyle("-fx-background-image: url('file:C:/Users/srdam/Documents/NetBeansProjects/BankUsersGUI/show.png');");
    } else {
        passT.setText(showpassT.getText());
        passT.setVisible(true);
        showpassT.setVisible(false);
        showB.setStyle("-fx-background-image: url('file:C:/Users/srdam/Documents/NetBeansProjects/BankUsersGUI/show.png');");
    }
}

```

```

@FXML
private void ok(ActionEvent event) throws SQLException, IOException {
    if (chek()) {
        SettingController.s = DB.getcolor(cardnumber);
        SettingController.lang = DB.getlang(cardnumber);
        scene.changescene("/chois/chois.fxml", event);
        if (Remember.isSelected()) {
            remember.write(cardT.getText(), passT.getText());
        } else {
            remember.write("", "");
        }
    }
}

public Boolean chek() throws SQLException {
    Boolean a;
    ResultSet rs;
    try {
        rs = DB.signin(cardT.getText(), passT.getText());
        if (rs.getRow() == 1) {
            a = true;
            incorrect.setVisible(false);
            cardnumber = rs.getInt("cardnumber");
        } else {
            a = false;
            incorrect.setVisible(true);
        }
    } catch (Exception e) {
        a = false;
        incorrect.setVisible(true);
    }
    return a;
}

```

هنا نتابع في كلاس SigninController وهنا يوجد الدوال التالية

١- changebutton والتي تقوم بتغيير style الزر عند مرور الفأرة عليه ليصبح لونه أزرق وتحت خط وهو ينفذ مع الزرين forget و signup

٢- returnbutton والتي تعيد الزر إلى حاله الطبيعية بعد خروج الفأرة من حدوده وهي تنفذ مع الزرين forget و signup

٣- show والتي تقوم بإظهار كلمة السر وإخفاءها وتغيير صورة الزر showB

٤- chek والتي تتأكد إذا كان رقم البطاقة وكلمة السر المدخلة صحيحة أو خاطئة وترجع boolean يمثل ذلك

٥- ok إذا كانت chek ترجع true تقوم بتعديل قيمة المتغير s من الكلاس settingController والمتغير lang من نفس الكلاس إلى قيم من ال database وتتأكد من

الزر rememberme إذا كان محددا فتقوم بكتابة رقم البطاقة وكلمة السر في الملف rememberme وإذا كان غير محدد تقوم بكتابة المحرف الصفر في أول

سطين

```

public class SendEmail {

    final String senderEmailID = "mohamadhallakbank@gmail.com";
    final String senderPassword = "mohamad2002";
    final String emailSMTPserver = "smtp.gmail.com";
    final String emailServerPort = "465";
    String receiverEmailID = "";
    static String emailSubject = "Test Mail";
    static String emailBody = ":";

    public SendEmail(String receiverEmailID, String Subject, String Body) {
        this.receiverEmailID = receiverEmailID;
        this.emailSubject = Subject;
        this.emailBody = Body;
        Properties props = new Properties();
        props.put("mail.smtp.user", senderEmailID);
        props.put("mail.smtp.host", emailSMTPserver);
        props.put("mail.smtp.port", emailServerPort);
        props.put("mail.smtp.starttls.enable", "true");
        props.put("mail.smtp.auth", "true");
        props.put("mail.smtp.socketFactory.port", emailServerPort);
        props.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
        props.put("mail.smtp.socketFactory.fallback", "false");
        SecurityManager security = System.getSecurityManager();
        try {
            Authenticator auth = new SMTPAuthenticator();
            Session session = Session.getInstance(props, auth);
            MimeMessage msg = new MimeMessage(session);
            msg.setText(emailBody);
            msg.setSubject(emailSubject);
            msg.setFrom(new InternetAddress(senderEmailID));
            msg.addRecipient(Message.RecipientType.TO,
                new InternetAddress(receiverEmailID));
            Transport.send(msg);
        } catch (Exception mex) {
            mex.printStackTrace();
        }
    }

    public class SMTPAuthenticator extends javax.mail.Authenticator {

        public PasswordAuthentication getPasswordAuthentication() {
            return new PasswordAuthentication(senderEmailID, senderPassword);
        }
    }
}

```

```

public String dialogSplit() throws IOException {
    String confirm;
    TextInputDialog dialog = new TextInputDialog();
    if (SettingController.lang) {
        dialog.setTitle("إنشاء حساب");
        dialog.setHeaderText("أدخل الرمز الذي تم إرساله إلى بريدك الإلكتروني");
    } else {
        dialog.setTitle("signup");
        dialog.setHeaderText("input the code that was sended to your email");
    }
    Optional<String> result = dialog.showAndWait();
    if (result.isPresent()) {
        confirm = result.get();
    } else {
        confirm = null;
    }
    return confirm;
}

```

هنا في الكلاس SendEmail دالة تنشئ كائن من الكلاس TextInputDialog والذي هو نافذة يتم فيها طلب الرمز الذي تم إرساله إلى الحساب وتقوم بإرجاع String يحمل قيمة الـ text المدخل

```

package signin;

import java.io.File;
import java.io.IOException;
import javaapplication37.mohamadBuffered;

public class signinmethods {
    private mohamadBuffered BFW=null;
    private File file=null;
    public void write(String a, String b ) throws IOException {
        file=new File("remember.txt");
        file.createNewFile();
        BFW=new mohamadBuffered(file);
        BFW.write(a,b);
    }
}

```

الكلاس signinmethods فيه الدالة write التي تأخذ في بارامترات متغيرين String وتقوم بكتابة هذين المتغيرين في الملف remember

هنا الكلاس SendEmail والذي يقوم بإرسال رسالة من ويأخذ في الكونستركت ثلاث متغيرات String الأول هو البريد الإلكتروني للمرسل إليه والثاني موضوع الرسالة والثالث محتوى الرسالة



```
public class ForgetController implements Initializable {

    @FXML
    private AnchorPane pane;
    @FXML
    private Label incorrect;
    @FXML
    private JFXTextField cardT;
    @FXML
    private JFXToggleButton choi;
    @FXML
    private JFXTextField confirm;
    @FXML
    private JFXButton ok;
    @FXML
    private JFXButton back;
    @FXML
    private Label pass;

    DB DB = new DB();
    ResultSet rs = null;
    Scens scene = new Scens();

    @Override
    public void initialize(URL url, ResourceBundle rb) {
        pane.setStyle(s);
        scene.scene("/signin/signin.fxml", back);
        if (SettingController.lang) {
            cardT.setPromptText("رقم البطاقة");
            back.setText("رجوع");
            ok.setText("انطلق");
            choi.setText("استخدام البريد الإلكتروني");
            confirm.setPromptText("البريد الإلكتروني");
            incorrect.setText("خطأ في رقم البطاقة أو البريد الإلكتروني");
        }
    }
}
```

هنا الواجهة الثالثة forget المرتبطة بالكلاس forgetController والذي فيه يتم ربط عناصر الواجهة بمتغيرات ثم تعريف متغيرات من كلاسات أخرى وتعريف resultSet ثم الدالة initialize والتي فيها يتم تعديل الstyle الخاص بمدير التنسيق ووضع قيمة الcss من المتغير s من الكلاس SettingController وثم استدعاء الدالة scene لتغيير الواجهة عند الضغط على زر الback ثم تغيير النص اعتمادا على اللغة المحددة من المتغير lang من الكلاس SettingController

card number

by email

email

back

OK

```

@FXML
void OK(ActionEvent event) throws SQLException {
    try {
        if (!chois.isSelected()) {
            rs = DB.forgetemail(cardT.getText(), confirm.getText());
            if (rs.getRow() == 1) {
                if (SettingController.lang) {
                    SendEmail a = new SendEmail(confirm.getText(), "استرجاع كلمة المرور", "كلمة مرورك هي: " + rs.getString("password"));
                    pass.setText("تم إرسال رسالة إلى بريدك الإلكتروني تحوي كلمة سر حسابك");
                } else {
                    SendEmail a = new SendEmail(confirm.getText(), "Password Recovery", "your password is: " + rs.getString("password"));
                    pass.setText("we have send your password to your email");
                }
                confirm.clear();
                cardT.clear();
                incorrect.setVisible(false);
            } else {
                incorrect.setVisible(true);
                pass.setText("");
            }
        } else if (chois.isSelected()) {
            rs = DB.forgetphone(cardT.getText(), confirm.getText());
            if (rs.getRow() == 1) {
                if (SettingController.lang) {
                    pass.setText("كلمة المرور هي: " + rs.getString("password"));
                } else {
                    pass.setText("your password is: " + rs.getString("password"));
                }
                incorrect.setVisible(false);
            } else {
                incorrect.setVisible(true);
            }
        }
    } catch (Exception e) {
        incorrect.setVisible(true);
        pass.setText("");
    }
}

@FXML
private void toggle(ActionEvent event) {
    if (chois.isSelected()) {
        if (SettingController.lang) {
            chois.setText("استخدام رقم الهاتف");
            confirm.setPromptText("رقم الهاتف");
            incorrect.setText("خطأ في رقم الهاتف أو رقم البطاقة");
        } else {
            chois.setText("By phonenumber");
            confirm.setPromptText("onenumber");
            incorrect.setText("incorrect phonenumber or cardnumber");
        }
    } else if (!chois.isSelected()) {
        if (SettingController.lang) {
            chois.setText("استخدام البريد الإلكتروني");
            confirm.setPromptText("البريد الإلكتروني");
            incorrect.setText("خطأ في رقم البطاقة أو البريد الإلكتروني");
        } else {
            chois.setText("By email");
            confirm.setPromptText("email");
            incorrect.setText("incorrect card number or email");
        }
    }
    confirm.setText("");
    incorrect.setVisible(false);
}

```

الدالة ok عند النقر على الزر ok فهي ستأكد من الchois إذا كان محددا وكانت المدخلات صحيحة سترسل كلمة المرور برسالة إلى البريد الإلكتروني المدخل وإذا كان chois غير محدد ستأكد من المدخلات وتضع كلمة السر في الlabel pass

Activate Wind
Go to Settings to a

الدالة toggle تقوم بتغيير نص الchois وتلميح الconfirm ونص الincorrect بحسب الاختيار واللغة

Full name

email

password

confirm password

phone number

birthday

☒ mail
☐ Femail

back

OK

```

@FXML
private AnchorPane pane;
@FXML
private JFXTextField name;
@FXML
private JFXTextField email;
@FXML
private JFXPasswordField confirm;
@FXML
private JFXTextField phone;
@FXML
private JFXPasswordField pass;
@FXML
private JFXDatePicker date;
@FXML
private JFXRadioButton male;
@FXML
private ToggleGroup TG;
@FXML
private JFXRadioButton female;
@FXML
private JFXButton ok;
@FXML
private JFXButton back;
@FXML
private Label namefalse;
@FXML
private Label emailfalse;
@FXML
private Label passfalse;
@FXML
private Label confalse;
@FXML
private Label phonefalse;
@FXML
private Label datefalse;
@FXML
private JFXTextField showpass;
@FXML
private JFXTextField showconfirm;
@FXML
private JFXButton showB;

```

```

DB DB = new DB();
Integer random = (int) (Math.random() * 100000);
Scens scene = new Scens();
@Override
public void initialize(URL url, ResourceBundle rb) {
    pane.setStyle(s);
    scene.scene("/signin/signin.fxml", back);
    showB.setStyle("-fx-background-image: url('file:C:/Users/srdam/Documents/NetBeansProjects/BankUsersGUI/show.png');-fx-background-size:cover");
    if (SettingController.lang) {
        name.setPromptText("الاسم الكامل");
        email.setPromptText("البريد الإلكتروني");
        pass.setPromptText("كلمة المرور");
        confirm.setPromptText("تأكيد كلمة المرور");
        phone.setPromptText("رقم الهاتف");
        date.setPromptText("تاريخ الميلاد");
        male.setText("ذكر");
        female.setText("أنثى");
        ok.setText("انطلق");
        back.setText("رجوع");
        namefalse.setText("يجب عليك إدخال اسم");
        emailfalse.setText("بريد الكتروني غير صحيح");
        passfalse.setText("حرفا صغيرا وحرفا كبيرا ورموزحرف من الحروف الخاصة");
        confalse.setText("لا يوجد تطابق");
        phonefalse.setText("رقم هاتف غير صحيح");
        datefalse.setText("يجب عليك إدخال تاريخ ميلاد");
    }
}

@FXML
private void show(ActionEvent e) {
    if (pass.isVisible()) {
        showpass.setText(pass.getText());
        showconfirm.setText(confirm.getText());
        pass.setVisible(false);
        confirm.setVisible(false);
        showpass.setVisible(true);
        showconfirm.setVisible(true);
        showB.setStyle("-fx-background-image: url('file:C:/Users/srdam/Documents/NetBeansProjects/BankUsersGUI/show.png');-fx-background-size:cover");
    } else {
        pass.setText(showpass.getText());
        confirm.setText(showconfirm.getText());
        pass.setVisible(true);
        confirm.setVisible(true);
        showpass.setVisible(false);
        showconfirm.setVisible(false);
        showB.setStyle("-fx-background-image: url('file:C:/Users/srdam/Documents/NetBeansProjects/BankUsersGUI/show.png');-fx-background-size:cover");
    }
}

```

Activate
 Go to Setting

هنا الواجهة الرابعة signup المرتبطة بالكلاس SignupController والذي فيه نبدأ بربط المتغيرات مع عناصر الواجهة ثم تعريف متغيرات من كلاسات أخرى وتعريف كائن نوعه integer قيمته من الكلاس math. Random وهو الذي سيتم إرساله إلى البريد الإلكتروني كرمز تأكيد ثم الدالة initialize والتي فيها نقوم بتحديد style لمدير التنسيق من المتغير s من الكلاس SettingController ثم استدعاء الدالة scene للتغير الواجهة عند النقر على زر back وإضافة صورة إلى الزر showB من خلال الstyle وتغير النص إذا كانت لغة البرنامج عربية ويتم تحديد هذا من المتغير lang من الكلاس SettingController ثم دالة show وهي مرتبطة بالزر showB وهي تقوم بإخفاء وإظهار كلمة السر مع تغير الصورة

Full name

email

password

confirm password

phone number

birthday

☒ mail
☐ Femail

back

OK

```

@FXML
private void ok(ActionEvent event) throws SQLException, IOException {
    if (!pass.isVisible()) {
        pass.setText(showpass.getText());
        confirm.setText(showconfirm.getText());
    }
    Boolean a, b, c, d, e, f;
    Boolean sex;
    if (name.getText() == null || "".equals(name.getText())) {
        a = false;
        namefalse.setVisible(true);
        name.setStyle("-fx-text-fill:red");
    } else {
        a = true;
        namefalse.setVisible(false);
        name.setStyle("-fx-text-fill:black");
    }
    if (email.getText().length() > 7 && email.getText().contains("@") && ".com".equals(email.getText().substring(email.getText().length() - 4, email.getText().length()))) {
        b = true;
        emailfalse.setVisible(false);
        email.setStyle("-fx-text-fill:black");
    } else {
        b = false;
        emailfalse.setVisible(true);
        email.setStyle("-fx-text-fill:red");
    }
    if (scene.checkpass(pass.getText()) == 5) {
        c = true;
        passfalse.setVisible(false);
        pass.setStyle("-fx-text-fill:black");
        showpass.setStyle("-fx-text-fill:black");
    } else {
        c = false;
        passfalse.setVisible(true);
        pass.setStyle("-fx-text-fill:red");
        showpass.setStyle("-fx-text-fill:red");
    }
    if (confirm.getText() == null ? pass.getText() == null : confirm.getText().equals(pass.getText())) {
        d = true;
        confirmfalse.setVisible(false);
        confirm.setStyle("-fx-text-fill:black");
        showconfirm.setStyle("-fx-text-fill:black");
    } else {
        d = false;
        confirmfalse.setVisible(true);
        confirm.setStyle("-fx-text-fill:red");
        showconfirm.setStyle("-fx-text-fill:red");
    }
    if (phone.getText().length() == 10 && phone.getText().charAt(0) == '0') {
        try {
            Integer.parseInt(phone.getText());
            e = true;
            phonefalse.setVisible(false);
            phone.setStyle("-fx-text-fill:black");
        } catch (Exception ex) {
            e = false;
            phonefalse.setVisible(true);
            phone.setStyle("-fx-text-fill:red");
        }
    } else {
        e = false;
        phonefalse.setVisible(true);
        phone.setStyle("-fx-text-fill:red");
    }
    if (date.getValue() == null) {
        f = false;
        datefalse.setVisible(true);
    } else {
        f = true;
        datefalse.setVisible(false);
    }
    sex = male.isSelected();
}

```

هنا نتابع في كلاس SignupController
وهنا الدالة ok التي تنفذ عند الضغط على
الزر ok وفيها هنا عبارة عن شروط
للمدخلات

١- الاسم: يجب أن لا يكون فارغا
٢- البريد الإلكتروني: يجب أن يكون طوله
أكثر من ٧ أحرف ويحتوي @ وفي نهايته
.com

٣- كلمة المرور: يجب أن تحقق شروط
الدالة chekpass من الكلاس Scens التي
تكلما عنها سابقا


٤- تأكيد كلمة المرور: يجب أن تماثل كلمة
المرور

٥- رقم الهاتف: يجب أن يكون طوله ١٠
أحرف وأن يبدأ ب0

٦-تاريخ الميلاد: يجب تحديد تاريخ ميلاد
وعدم تركه فارغا


Full name

email

password 

confirm password

phone number

birthday 

☒ mail ☐ Femail

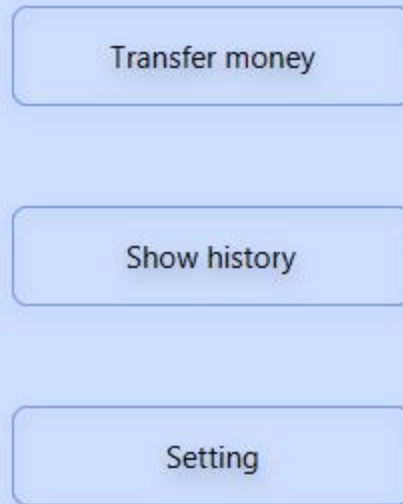
back OK

```

sex = male.isSelected();
if (a == b == c == d == e == f) {
    SendEmail send = new SendEmail(email.getText(), "confirm Email", randomD.toString());
    String confirm = send.dialogSplit();
    if (confirm.equals(randomD.toString())) {
        DB.add(name.getText(), email.getText(), pass.getText(), phone.getText(), Date.valueOf(date.getValue()), sex);
        Alert yourcardnum = new Alert(Alert.AlertType.INFORMATION);
        if (SettingController.lang) {
            yourcardnum.setTitle("رقم البطاقة");
            yourcardnum.setContentText("رقم البطاقة الخاصة بك هو: " + DB.getnewCN());
        } else {
            yourcardnum.setTitle("Card Number");
            yourcardnum.setContentText("Your card number is: " + DB.getnewCN());
        }
        yourcardnum.setHeaderText("");
        yourcardnum.getDialogPane().setStyle("-fx-background-color:#ccddff;-fx-font-size:15");
        yourcardnum.showAndWait();
    } else {
        Alert wrong = new Alert(Alert.AlertType.ERROR);
        if (SettingController.lang) {
            wrong.setTitle("خطأ");
            wrong.setContentText("خطأ في الرمز");
        } else {
            wrong.setTitle("incorrect");
            wrong.setContentText("wrong confirm code");
        }
        wrong.setHeaderText("");
        wrong.getDialogPane().setStyle("-fx-background-color:#ccddff;-fx-font-size:15");
        wrong.showAndWait();
    }
}
}

```

هنا نتابع في كلاس SignupController في الدالة Ok بعد التحقق من شروط المدخلات يتم إرسال رمز التأكيد إلى البريد الإلكتروني المدخل ثم تظهر نافذة تطلب هذا الرمز فإذا كان صحيحا يتم إضافة المستخدم إلى ال database وتظهر نافذة تخبر المستخدم برقم بطاقته وإذا لم يكن صحيحا تظهر نافذة تخبره بالخطأ الموجود



log out

```
public class ChoisController implements Initializable {  
    @FXML  
    private JFXButton transfer;  
    @FXML  
    private JFXButton show;  
    @FXML  
    private JFXButton Setting;  
    @FXML  
    private JFXButton logout;  
    @FXML  
    private AnchorPane pane;  
  
    Scens scene = new Scens();  
    signinmethods remember = new signinmethods();  
  
    @Override  
    public void initialize(URL url, ResourceBundle rb) {  
        pane.setStyle(s);  
        scene.scene("/transfer/transfer.fxml", transfer);  
        scene.scene("/show/show.fxml", show);  
        scene.scene("/Setting/setting.fxml", Setting);  
        logout(logout);  
        if (SettingController.lang) {  
            transfer.setText("تحويل مال");  
            show.setText("تفقد المعاملات");  
            Setting.setText("إعدادات");  
            logout.setText("تسجيل خروج");  
        }  
    }  
  
    public void logout(JFXButton a) {  
        a.setOnAction((e) -> {  
            try {  
                remember.write("", "");  
                scene.changescene("/signin/signin.fxml", e);  
            } catch (IOException ex) {  
                System.out.println(ex.getMessage());  
            }  
        }  
    }  
}
```

هنا الواجهة الخامسة chois المرتبطة بالكلاس choisController والذي فيه ربط المتغيرات أولا ثم استدعاء متغيرات من كلاسات أخرى

الدالة logout: تأخذ في باراميتراتها متغير واحد نوعه button وتضيف event لهذا الزر في هذا الevent يتم كتابة المحرف الصفري في الملف remember كاسم مستخدم وكلمة مرور وتبديل الواجهة إلى واجهة SigninController باستعمال الدالة changescene من الكلاس Scens

ثم دالة initialize والتي يتم فيها تحديد style لمدير التنسيق قيمته من المتغير s من الكلاس SettingController واستدعاء دالة scene من الكلاس Scens التي تقوم بالتبديل بين الواجهات حسب الزر المنقور عليه واستدعاء الدالة logout وتغير النص بحسب اللغة ويتم تحديد اللغة من المتغير lang من الكلاس SettingControlle

log out

To Who (card number)

how much

back

OK

```

public class TransferController implements Initializable {
    @FXML
    private AnchorPane pane;
    @FXML
    private JFXTextField who;
    @FXML
    private JFXSlider moneyS;
    @FXML
    private JFXTextField moneyT;
    @FXML
    private JFXButton logout;
    @FXML
    private JFXButton ok;
    @FXML
    private JFXButton back;

    Scens scene = new Scens();
    DB DB = new DB();
    ChoisController clear = new ChoisController();

    @Override
    public void initialize(URL url, ResourceBundle rb) {
        try {
            moneyS.setMax(DB.getbalance(SigninController.cardnumber.toString()));
        } catch (SQLException ex) {
            System.out.println(ex.getMessage());
        }
        pane.setStyle(s);
        scene.scene("/chois/chois.fxml", back);
        NumberValidator nu = new NumberValidator("you have to input just numbers");
        who.getValidators().add(nu);

        who.focusedProperty().addListener((ObservableValue extends Boolean observable, Boolean oldValue, Boolean newValue)
            if (!newValue) {
                who.validate();
            }
        });
        clear.logout(logout);
        if (SettingController.lang) {
            back.setText("رجوع");
            ok.setText("انطلق");
            logout.setText("تسجيل خروج");
            who.setPromptText("رقم البطاقة لمن");
        }
        moneyT.focusedProperty().addListener((ObservableValue extends Boolean observable, Boolean oldValue, Boolean newVal
        Boolean bol;
        try {
            Double.parseDouble(moneyT.getText());
            bol = true;
        } catch (Exception ex) {
            bol = false;
        }
        if (bol && moneyT.getText() != null && !"".equals(moneyT.getText())) {
            if (Double.parseDouble(moneyT.getText()) >= moneyS.getMax()) {
                moneyT.setText(String.valueOf(moneyS.getMax()));
            }
            moneyS.setValue(Double.parseDouble(moneyT.getText()));
        } else {
            moneyS.setValue(0.0);
        }
    }
}

```

هنا الواجهة السادسة المرتبطة بالكلاس TransferController الذي فيه :

نربط عناصر الواجهة بمتغيرات ومعرف عناصر من كلاسات أخرى ثم دالة initialize التي فيها ضع القيمة العظمى لل slider من ال database ونضع style لمدير التنسيق من المتغير s من الكلاس SettingController ونستدعي دالة scene للتغير الواجهة ونضيف شرط لإدخال ال cardnumber كرقم فقط ثم نستدعي دالة logout لتسجيل الخروج ونغير النص حسب اللغة التي يتم تحديدها من المتغير lang

ثم نضيف حدث ل moneyT عند تحديده أو إزالة التحديد عنه يتم تغير قيمة ال Slider حسب القيمة المدخلة

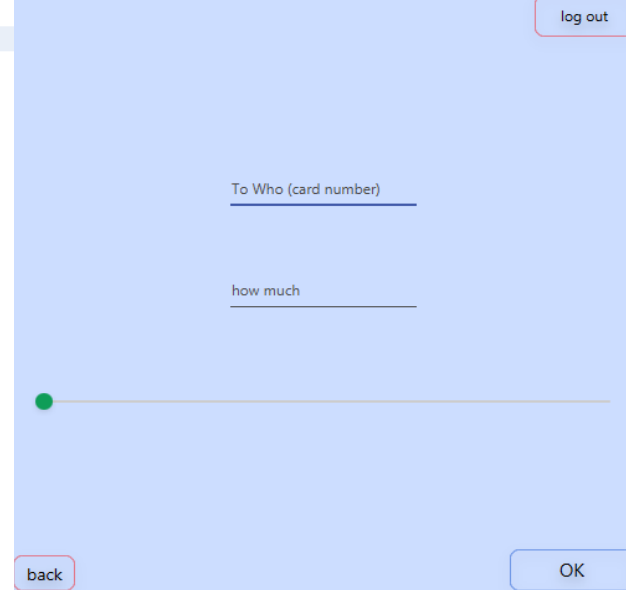
```

@FXML
private void ok(ActionEvent event) throws SQLException {
    Boolean b;
    try {
        Integer.parseInt(who.getText());
        ResultSet rs = DB.getpersonal(Integer.parseInt(who.getText()));
        b = rs.getRow() == 1;
    } catch (NumberFormatException | SQLException e) {
        b = false;
    }
    Alert moneytransferred;
    if (b) {
        Double moneyvalue = this.moneyS.getValue();
        Alert chektransfer = new Alert(Alert.AlertType.CONFIRMATION);
        if (who.getText().equals(SigninController.cardnumber.toString())) {
            moneytransferred = new Alert(Alert.AlertType.ERROR);
            if (SettingController.lang) {
                moneytransferred.setTitle("تحويل مال");
                moneytransferred.setContentText("لا يمكنك تحويل مال لحسابك");
            } else {
                moneytransferred.setTitle("transfer money");
                moneytransferred.setContentText("you can't transfer money to yourself");
            }
            moneytransferred.setHeaderText("");
            moneytransferred.getDialogPane().setStyle("--fx-background-color:#ccddff;--fx-font-size:15");
            moneytransferred.showAndWait();
        } else {
        } else if (option.get() == ButtonType.CANCEL) {

        } else {

        }
    }
} else {
    moneytransferred = new Alert(Alert.AlertType.ERROR);
    if (SettingController.lang) {
        moneytransferred.setTitle("تحويل مال");
        moneytransferred.setContentText("لا يوجد حساب يحمل رقم البطاقة المدخل");
    } else {
        moneytransferred.setTitle("Transfer money");
        moneytransferred.setContentText("there are no account with this cardnumber");
    }
    moneytransferred.setHeaderText("");
    moneytransferred.getDialogPane().setStyle("--fx-background-color:#ccddff;--fx-font-size:15");
    moneytransferred.showAndWait();
}
}

```



```

ChangeListener CL = new ChangeListener<Number>() {
    @Override
    public void changed(ObservableValue observable, Number oldValue, Number newValue) {
        if (newValue != null) {
            moneyT.setText(newValue.toString().substring(0, newValue.toString().indexOf(".") + 2));
        } else {
            moneyT.setText("0.0");
        }
    }
};

moneyS.setOnMousePressed((e) -> {
    moneyS.valueProperty().addListener(CL);
    moneyT.setText(String.valueOf(moneyS.getValue()).substring(0, String.valueOf(moneyS.getValue()).indexOf(".") + 2));
});
moneyS.setOnMouseReleased((e) -> {
    moneyS.valueProperty().removeListener(CL);
});

```

وهنا الدالة ok المرتبطة مع الزر OK تقوم بالتأكد من رقم البطاقة المدخل إذا كان هناك مستخدم يحمله أو لا وإذا كان هو رقم البطاقة للمستخدم الحالي وفي كل حالة تظهر alert تحمل رسالة خطأ

هنا ما زلنا في كلاس TransferController في دالة initialize حيث نقوم بإضافة changListeren يتم فيها وضع نص في moneyT حسب الSlider ويتم استدعائها في عند النقر على الslider ويتم إزالتها عند إزالة النقر عنه

log out

To Who (card number)

how much

back

OK

```

else {
    if (SettingController.lang) {
        chektransfer.setTitle("تحويل مال");
        chektransfer.setContentText("هل أنت متأكد من تحويل مبلغ قدره:" + moneyvalue.toString().substring(0, moneyvalue.toString().length()));
    } else {
        chektransfer.setTitle("Transfer money");
        chektransfer.setContentText("are you sure you want to transfer:" + moneyvalue.toString().substring(0, moneyvalue.toString().length()));
    }
    chektransfer.setHeaderText("");
    chektransfer.getDialogPane().setStyle("-fx-background-color:#ccddff;-fx-font-size:15");
    Optional<ButtonType> option = chektransfer.showAndWait();

    if (option.get() == null) {

    } else if (option.get() == ButtonType.OK) {
        DB.transfer(who.getText(), moneyvalue);
        who.setText("");
        moneytransferred = new Alert(Alert.AlertType.INFORMATION);
        if (SettingController.lang) {
            moneytransferred.setTitle("تحويل مال");
            moneytransferred.setContentText("تم تحويل المال");
        } else {
            moneytransferred.setTitle("Transfer money");
            moneytransferred.setContentText("Transfer compleat");
        }
        moneytransferred.setHeaderText("");
        moneytransferred.getDialogPane().setStyle("-fx-background-color:#ccddff;-fx-font-size:15");
        moneyS.setMax(DB.getbalance(SigninController.cardnumber.toString()));
        moneyS.setValue(1);
        moneyT.setText("1.0");
        moneytransferred.showAndWait();
    }
}

```

هنا نتابع في كلاس TransferController في دالة ok بعد التأكد من رقم البطاقة المدخل تقوم بإظهار Alert للتأكد من عملية النقل فإذا تم التأكيد تقوم بإضافة بيانات هذه العملية إلى الجدول money من ال database وتظهر رسالة لتعلم المستخدم بتمام العملية

[illegible][back](#)[log out](#)

show by name ▼

```

public class ShowController implements Initializable {

@FXML
private AnchorPane pane;
@FXML
private JFXButton logout;
@FXML
private JFXButton back;
@FXML
private TableView<User> TV;
@FXML
private ChoiceBox<String> choicebox;
@FXML
Label balance;

public static String chois = "name";
DB DB = new DB();
Scens scene = new Scens();
ObservableList<User> data = null;
ChoisController clear = new ChoisController();

@Override
public void initialize(URL url, ResourceBundle rb) {
    pane.setStyle(s);
    scene.scene("/chois/chois.fxml", back);
    clear.logout(logout);
    try {
        balance.setText("Your balance is: " + DB.getbalance(SigninController.cardnumber.toString()));
    } catch (SQLException ex) {
        System.out.println(ex.getMessage());
    }
}

if (SettingController.lang) {
    choicebox.getItems().addAll("إظهار باستعمال البريد الإلكتروني", "إظهار باستعمال البطاقة", "إظهار باستعمال الاسم");
    choicebox.setValue("show by name");
} else {
    choicebox.getItems().addAll("show by name", "show by email", "show by card number");
    choicebox.setValue("show by name");
}

choicebox.getSelectionModel().selectedIndexProperty().addListener(new ChangeListener<Number>() {

@Override
public void changed(ObservableValue<? extends Number> observable, Number oldValue, Number newValue) {
    switch ((Integer) newValue) {
        case 0:
            chois = "name";
            break;
        case 1:
            chois = "email";
            break;
        case 2:
            chois = "cardnumber";
            break;
    }
    try {
        data = DB.show();
    } catch (SQLException ex) {
        System.out.println(ex.getMessage());
    }
    TV.setItems(data);
}

try {
    data = DB.show();
} catch (SQLException ex) {
    System.out.println(ex.getMessage());
}
}
}

```

هنا الواجهة السابعة showhistory المرتبطة بالكلاس
showController الذي فيه نبدأ بربط عناصر الواجهة
بمتغيرات من الكلاس ثم استدعاء متغيرات من كلاسات
أخرى وتعريف observableList تأخذ قيم نوعها user
الكلاس User ثم الدالة initialize حيث فيها نقوم بتحديد
style لمدير التنسيق من المتغير s من الكلاس
SettingController ونستدعي الدالة scene لتبديل الوا
عند الضغط على زر back ودالة logout لتسجيل الخرو
ثم إضافة نص إلى balance يحمل قيمة رصيد المستخدم
إضافة عناصر للchoicebox حسب اللغة وإضافة حدث
يقوم بتغير قيمة المتغير choi الذي يحدد طريقة عرض
البيانات في الجدول

هنا نتابع في الكلاس

showController في الدالة

initialize نضيف هنا أعمدة إلى

الجدول وبيانات ونحدد عرض كل

واحدة منها

وهنا الكلاس User له هو لكلاس الخاص

بيانات الجدول وفيه كونستركتران

ودالتين get و set لكل متغير فيه

[illegible][log out](#)

show by name ▼

```
TableColumn<User, String> columnmoney = new TableColumn<>("money");
columnmoney.setCellValueFactory(new PropertyValueFactory<>("Money"));
```

```
TableColumn<User, String> columnFrom = new TableColumn<>("From");
columnFrom.setCellValueFactory(new PropertyValueFactory<>("From"));
```

```
TableColumn<User, String> columnTo = new TableColumn<>("To");
columnTo.setCellValueFactory(new PropertyValueFactory<>("To"));
```

```
TableColumn<User, String> columndate = new TableColumn<>("Date");
columndate.setCellValueFactory(new PropertyValueFactory<>("Date"));
```

```
TV.setColumns().addAll(columnmoney, columnFrom, columnTo, columndate);
```

```
TV.setItems(data);
```

```
columnmoney.setPrefWidth(100);
```

```
columnFrom.setPrefWidth(100);
```

```
columnTo.setPrefWidth(100);
```

```
columndate.setPrefWidth(100);
```

}

```
public class User {
```

```
SimpleStringProperty Money;
```

```
SimpleStringProperty From;
```

```
SimpleStringProperty To;
```

```
SimpleStringProperty Date;
```

```
public User() {
```

```
this.Money = new SimpleStringProperty("");
```

```
this.From = new SimpleStringProperty("");
```

```
this.To = new SimpleStringProperty("");
```

```
this.Date = new SimpleStringProperty("");
```

}

```
public User(String money, String from, String to,String date) {
```

```
this.Money = new SimpleStringProperty(money);
```

```
this.From = new SimpleStringProperty(from);
```

```
this.To = new SimpleStringProperty(to);
```

```
this.Date = new SimpleStringProperty(date);
```

}

```
public String getMoney() {
```

```
return Money.getValue();
```

}

```
public void setMoney (String Money) {
```

```
this.Money = new SimpleStringProperty(Money);
```

}

[back](#)



هنا الواجهة الثامنة Setting المرتبطة بالكلاس SettingController والذي فيه أولا ربط عناصر الواجهة بمتغيرات الكلاس وتعريف كائنات من كلاسات أخرى وتعريف المتغيرين lang,s وهما المتغيران اللذان يحددان لغة الواجهة ولونها وهما static ليتم تغييرهما من باقي الكلاسات والمحافظة على تغييرهما ثم دالة initialize والتي فيها أولا إضافة style لمدير التنسيق من المتغير s وإضافة صورة للزر showB الذي يستعمل لإظهار وإخفاء كلمة المرور واستدعاء الدالة scene لتغيير الواجهة عند الضغط على زر back والدالة logout لتسجيل الخروج ثم تعيين إذا كان الToggleButton true or false وذلك تبعاً للمتغير lang ثم استدعاء دالة changeL التي تغير النص تبعاً للمتغير lang والدالة changecolor وهي تقوم بتغيير لون الواجهة تبعاً للعنصر color الذي نوعه JFXColorPicker والذي يستعمل لتحديد لون ثم تعديل قيمة المتغير S ورفع القيمة إلى الdatabase و changeL هي دالة مرتبطة مع الLanguage ToggleButton والتي تغير النص في الواجهة تبعاً لتحديد هذا العنصر أو عدم تحديده وتعديل قيمة المتغير lang وترفع هذه القيمة إلى الdatabase

Background Color

#FFFFFF

Edit Personal Information

password

ok

english

log out

back

```
@FXML
private void chekpass(ActionEvent e) throws SQLException, IOException {
    ResultSet rs = DB.getpersonal();
    if (rs.getString("password").equals(password.getText())) {
        scene.changescene("/edit/edit.fxml", e);
        wrong.setVisible(false);
    } else {
        wrong.setVisible(true);
    }
}
```

@FXML

```
private void editevent(ActionEvent e) {
    password.setVisible(true);
    ok.setVisible(true);
}
```

@FXML

```
private void show(ActionEvent e) {
    if (password.isVisible()) {
        showpass.setText(password.getText());
        password.setVisible(false);
        showpass.setVisible(true);
        showB.setStyle("-fx-background-image: url('file:C:/Users/srdam/Documents/NetBeansProjects/BankUsersGUI/hide.png');");
    } else {
        password.setText(showpass.getText());
        password.setVisible(true);
        showpass.setVisible(false);
        showB.setStyle("-fx-background-image: url('file:C:/Users/srdam/Documents/NetBeansProjects/BankUsersGUI/show.png');");
    }
}
```

Activate Windows
Go to Settings to activate Windows

هنا نتابع في كلاس SettingController وهنا يوجد الدوال الأتية:

١- chekpass وهي دالة مرتبطة بالعنصر ok والتي تتأكد من كلمة المرور فإذا كانت صحيحة تغير الواجهة إلى edit وإلا تظهر الخطأ

٢- editevent وهي دالة مرتبطة بالزر edit والتي تقوم بإظهار العنصرين password و ok

٣- show وهي مرتبطة بالزر showB وهي تقوم بإظهار وإخفاء كلمة المرور وتغير الصورة الخاصة بالزر

back OK

ثم الدالة edit المرتبطة بكل الـ `RadioButton` الموجودة والتي تقوم بتفعيل خيار التعديل إذا كان الـ `RadioButton` محددًا وتلغي التفعيل عند إلغاء التحديد

ثم الدالة male المرتبطة بالmale ToggleButton والتي
تغير نص هذا العنصر تبعاً للغة وإذا كان محدداً أو لا

```

private void ok(ActionEvent event) throws SQLException {
    Boolean a, b, c, d;
    if (name.isEditable() && (name.getText() == null || "".equals(name.getText()))) {
        a = false;
        namefalse.setVisible(true);
        name.setStyle("-fx-text-fill:red");
    } else {
        a = true;
        name.setStyle("-fx-text-fill:black");
        namefalse.setVisible(false);
    }
    if ((pass.isEditable()) && !(scene.checkpass(pass.getText())==5)) {
        b = false;
        passfalse.setVisible(true);
        pass.setStyle("-fx-text-fill:red");
    } else {
        b = true;
        passfalse.setVisible(false);
        pass.setStyle("-fx-text-fill:black");
    }
    if ((confirm.isEditable()) && !(confirm.getText().equals(pass.getText()))) {
        c = false;
        confirmfalse.setVisible(true);
        confirm.setStyle("-fx-text-fill:red");
    } else {
        c = true;
        confirm.setStyle("-fx-text-fill:black");
        confirmfalse.setVisible(false);
    }
    if ((phone.isEditable()) && !(phone.getText().length() == 10 && phone.getText().charAt(0) == '0' && phone.getText().charAt(1) == '9')) {
        try {
            Integer.parseInt(phone.getText());
            d = false;
            phonefalse.setVisible(true);
            phone.setStyle("-fx-text-fill:red");
        } catch (Exception ex) {
            d = false;
            phonefalse.setVisible(true);
            phone.setStyle("-fx-text-fill:red");
        }
    } else {
        d = true;
        phonefalse.setVisible(false);
        phone.setStyle("-fx-text-fill:black");
    }
}

if (a && b && c && d) {
    DB.edit(name, pass, phone, date, mail);
    Alert editcomleat = new Alert(Alert.AlertType.INFORMATION);
    if (SettingController.lang) {
        editcomleat.setTitle("التعديل");
        editcomleat.setContentText("تم التعديل");
    } else {
        editcomleat.setTitle("edit");
        editcomleat.setContentText("edit compleat ");
    }
    editcomleat.setHeaderText("");
    editcomleat.getDialogPane().setStyle("-fx-background-color:#coddff;-fx-font-size:15");
    editcomleat.showAndWait();
}

private void edit(ActionEvent e) {
    if (e.getSource().equals(editname)) {
        name.setEditable(editname.isSelected());
    } else if (e.getSource().equals(editpass)) {
        pass.setEditable(editpass.isSelected());
        confirm.setEditable(editpass.isSelected());
    } else if (e.getSource().equals(editphone)) {
        phone.setEditable(editphone.isSelected());
    } else if (e.getSource().equals(editdate)) {
        date.setDisable(!editdate.isSelected());
    } else if (e.getSource().equals(editsex)) {
        mail.setDisable(!editsex.isSelected());
    }
}

private void mail(ActionEvent event) {
    if (mail.isSelected()) {
        if (SettingController.lang) {
            mail.setText("ذكر");
        } else {
            mail.setText("Male");
        }
    } else {
        if (SettingController.lang) {
            mail.setText("انثى");
        } else {
            mail.setText("Female");
        }
    }
}

```

في الدالة ok المرتبطة
 بروط المدخلات إذا كان خيار
 تجميعها صحيحة ترفع
 Alert تخبر المستخدم

RadioButtons الموجودة والتي
 محددا RadioButton

male ToggleButtons والتي
 كان محددا أو لا

```

public class DB {

    Connection conn = null;
    PreparedStatement stmt = null;
    ResultSet rs = null;
    PreparedStatement stmt1 = null;
    ResultSet rsl = null;

    public void conn() {
        try {
            SQLServerDataSource ds = new SQLServerDataSource();
            ds.setIntegratedSecurity(true);
            ds.setDatabaseName("users");
            conn = ds.getConnection();
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }

    public ResultSet signin(String s1, String s2) throws SQLException {
        conn();
        stmt = conn.prepareStatement("Select * From usersinformation WHERE cardnumber=? and password=?");
        stmt.setString(1, s1);
        stmt.setString(2, s2);
        rs = stmt.executeQuery();
        rs.next();
        return rs;
    }

    public ResultSet forgetemail(String s1, String s2) throws SQLException {
        conn();
        stmt = conn.prepareStatement("Select * From usersinformation WHERE cardnumber=? and email=?");
        stmt.setString(1, s1);
        stmt.setString(2, s2);
        rs = stmt.executeQuery();
        rs.next();
        return rs;
    }

    public ResultSet forgetphone(String s1, String s2) throws SQLException {
        conn();
        stmt = conn.prepareStatement("Select * From usersinformation WHERE cardnumber=? and phone=?");
        stmt.setString(1, s1);
        stmt.setString(2, s2);
        rs = stmt.executeQuery();
        rs.next();
        return rs;
    }

    public String getnewCN() throws SQLException {
        conn();
        String rns = "";
        stmt = conn.prepareStatement("Select * From usersinformation");
        rs = stmt.executeQuery();
        while (rs.next()) {
            rns = rs.getString("cardnumber");
        }
        return rns;
    }

    public Double getbalance(String card) throws SQLException {
        conn();
        stmt = conn.prepareStatement("Select * From usersinformation Where cardnumber=?");
        stmt.setString(1, card);
        rs = stmt.executeQuery();
        rs.next();
        return rs.getDouble("balance");
    }

    public ObservableList<User> show() throws SQLException {
        ObservableList<User> show = FXCollections.observableArrayList();
        conn();
        stmt = conn.prepareStatement("Select * FROM money WHERE [From]=? or [To]=?", ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);
        stmt.setInt(1, SigninController.cardnumber);
        stmt.setInt(2, SigninController.cardnumber);
        rs = stmt.executeQuery();
        rs.next();
        stmt1 = conn.prepareStatement("Select * From usersinformation WHERE cardnumber =?");
        if (rs.getRow() == 1) {
            rs.beforeFirst();
            while (rs.next()) {
                if (rs.getInt("From") == (SigninController.cardnumber)) {
                    stmt1.setString(1, rs.getString("TO"));
                    rsl = stmt1.executeQuery();
                    rsl.next();
                    show.add(new User("-" + rs.getString("money"), "YOU", rsl.getString>ShowController.chois), rs.getString("date")));
                } else if (rs.getInt("TO") == (SigninController.cardnumber)) {
                    stmt1.setString(1, rs.getString("From"));
                    rsl = stmt1.executeQuery();
                    rsl.next();
                    show.add(new User(rs.getString("money"), rsl.getString>ShowController.chois), "YOU", rs.getString("date")));
                }
            }
            rs.close();
            rsl.close();
            stmt.close();
            stmt1.close();
        }
        return show;
    }

    public ResultSet getpersonal() throws SQLException {
        conn();
        stmt = conn.prepareStatement("Select * From usersinformation Where cardnumber=?");
        stmt.setInt(1, SigninController.cardnumber);
        rs = stmt.executeQuery();
        rs.next();
        return rs;
    }

    public ResultSet getpersonal(int cardnumber) throws SQLException {
        conn();
        stmt = conn.prepareStatement("Select * From usersinformation Where cardnumber=?");
        stmt.setInt(1, cardnumber);
        rs = stmt.executeQuery();
        rs.next();
        return rs;
    }
}

```

هنا الكلاس DB والذي يمثل دوال ال database دواله هي : ١-conn وهي دالة إنشاء الاتصال ٢-signin وهي دالة للتأكد من صحة رقم البطاقة وكلمة لمروور عند تسجيل الدخول تأخذ باراميتران نوعهما String يمثلان رقم البطاقة وكلمة السر المراد التأكد منهما وترجع ResultSet ٣-forgetemail وهي دالة للتأكد من البريد الإلكتروني مع رقم البطاقة وتأخذ باراميتران نوعهما String يمثلان البريد الإلكتروني ورقم البطاقة وترجع ResultSet ٤-forgetphone وهي دالة مثل السابقة ولكن بدل البريد الإلكتروني رقم الهاتف ٥-getCN وهي دالة ترجع String يمثل رقم البطاقة للمستخدم الجديد ٦-getbalance وهي دالة تأخذ String كباراميتر يمثل رقم البطاقة المراد معرفة رصيدها وترجع double يمثل هذا الرصيد ٧- الدالة show والتي ترجع observableList تمثل بيانات المعاملات المالية للمستخدم الحالي

هنا نتابع في لكتاس DB

٨-editinfo والتي تقوم بتعديل المعلومات الشخصية وهي تأخذ

باراميتران نوعهما String الأول يمثل اسم العامود في

الdatabase الذي سيتم تعديل محتواه والثاني يمثل القيمة الجديدة

٩-editmale والتي تأخذ باراميتز واحد a وهو boolean وتقوم

بتعديل العامود male من الdatabase وتضع فيه القيمة a

١٠-edit والتي تأخذ في باراميتراتها ثلاث متغيرات a,b,c

نوعها JFXTextField ومتغير d نوعه JFXDatePicker ومتغير

e نوعه JFXToggleButton وتتأكد من كل واحد إذا كان قابل

للتعديل وتستدعي الدالة ٨ أو ٩ حسب المطلوب

١١-Transfare تأخذ باراميتران money نوعه Double يمثل

قيمة المال الذي تم تحويله و card نوعه String يمثل رقم بطاقة

الشخص المراد تحويل المال إليه وتقوم بإضافة معلومات التحويل

إلى الجدول money وتعديل رصيد المرسل والمرسل إليه

```
private void editinfo(String a, String b) throws SQLException {
    conn();
    stmt = conn.prepareStatement("Update usersinformation set " + a + " =? WHERE cardnumber=?", ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);
    stmt.setString(1, b);
    stmt.setInt(2, SigninController.cardnumber);
    stmt.executeUpdate();
    stmt.close();
    conn.close();
}

private void editmale(Boolean a) throws SQLException {
    conn();
    stmt = conn.prepareStatement("Update usersinformation set male =? WHERE cardnumber=?", ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);
    stmt.setBoolean(1, a);
    stmt.setInt(2, SigninController.cardnumber);
    stmt.executeUpdate();
    stmt.close();
    conn.close();
}

public void edit(JFXTextField a, JFXTextField b, JFXTextField c, JFXDatePicker d, JFXToggleButton e) throws SQLException {
    if (a.isEditable()) {
        editinfo("name", a.getText());
    }
    if (b.isEditable()) {
        editinfo("password", b.getText());
    }
    if (c.isEditable()) {
        editinfo("phone", c.getText());
    }
    if (!d.isDisable()) {
        editinfo("birthday", Date.valueOf(d.getValue()).toString());
    }
    if (!e.isDisable()) {
        editmale(e.isSelected());
    }
}

public void transfer(String card, Double money) throws SQLException {
    conn();
    stmt = conn.prepareStatement("INSERT INTO money " + "values (?, ?, ?, ?)");
    stmt.setInt(1, SigninController.cardnumber);
    stmt.setString(2, card);
    stmt.setDouble(3, money);
    stmt.setDate(4, Date.valueOf(LocalDate.now()));
    stmt.execute();
    conn.close();
    conn();
    try {
        SQLServerDataSource ds = new SQLServerDataSource();
        ds.setIntegratedSecurity(true);
        ds.setDatabaseName("users");
        conn = ds.getConnection();
        stmt1 = conn.prepareStatement("Update usersinformation set balance =? WHERE cardnumber=?", ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);
        stmt1.setDouble(1, getbalance(card) + money);
        stmt1.setString(2, card);
        Integer a = stmt1.executeUpdate();
    } catch (Exception e) {
        System.out.println(e.getMessage());
    } finally {
        if (rs != null) { try { rs.close(); } catch (Exception e) { System.out.println(e.getMessage()); } }
        if (stmt1 != null) { try { stmt1.close(); } catch (Exception e) { System.out.println(e.getMessage()); } }
        if (conn != null) { try { conn.close(); } catch (Exception e) { System.out.println(e.getMessage()); } }
    }
    try {
        SQLServerDataSource ds = new SQLServerDataSource();
        ds.setIntegratedSecurity(true);
        ds.setDatabaseName("users");
        conn = ds.getConnection();
        stmt1 = conn.prepareStatement("Update usersinformation set balance =? WHERE cardnumber=?", ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);
        stmt1.setDouble(1, getbalance(SigninController.cardnumber.toString()) - money);
        stmt1.setInt(2, SigninController.cardnumber);
        Integer a = stmt1.executeUpdate();
    } catch (Exception e) {
        System.out.println(e.getMessage());
    } finally {
        if (rs != null) { try { rs.close(); } catch (Exception e) { System.out.println(e.getMessage()); } }
        if (stmt1 != null) { try { stmt1.close(); } catch (Exception e) { System.out.println(e.getMessage()); } }
        if (conn != null) { try { conn.close(); } catch (Exception e) { System.out.println(e.getMessage()); } }
    }
}
```


١٢ - getColor وهي ترجع String يمثل قيمة العاود

color من database عندما رقم البطاقة يساوي

الباراميتر a الذي نوعه Integer الذي يمثل رقم البطاقة

١٣ - setColor وهي تقوم بتعديل قيمة العاود color إلى

قيمة الباراميتر a الذي نوعه String للمستخدم الحالي

١٤ - getlang و ١٥ - setlang هما دالتان كالدالتين ١٢

و ١٣ على الترتيب ولكن بدل اللون لغة

١٦ - add وهي دالة لإضافة مستخدم جديد في لجدول

usersinformation وتأخذ في باراميتراتها معلومات

المستخدم التي سيتم إضافتها إلى الجدول

```
public String getColor(Integer a) throws SQLException {
    String color;
    conn();
    stmt = conn.prepareStatement("Select * From usersinformation where cardnumber=?");
    stmt.setInt(1, a);
    rs = stmt.executeQuery();
    rs.next();
    if (rs.getString("color") != null) {
        color = rs.getString("color");
    } else {
        color = "-fx-background-color:#c0c0c0;";
    }
    return color;
}

public void setColor(String a) throws SQLException {
    conn();
    stmt = conn.prepareStatement("Update usersinformation set color =? WHERE cardnumber=?", ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);
    stmt.setString(1, a);
    stmt.setInt(2, SigninController.cardnumber);
    stmt.executeUpdate();
    stmt.close();
    conn.close();
}

public Boolean getlang(Integer a) throws SQLException {
    Boolean lang;
    conn();
    stmt = conn.prepareStatement("Select * From usersinformation where cardnumber=?");
    stmt.setInt(1, a);
    rs = stmt.executeQuery();
    rs.next();
    lang = rs.getBoolean("lang");
    return lang;
}

public void setlang(Boolean a) throws SQLException {
    conn();
    stmt = conn.prepareStatement("Update usersinformation set lang =? WHERE cardnumber=?", ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);
    stmt.setBoolean(1, a);
    stmt.setInt(2, SigninController.cardnumber);
    stmt.executeUpdate();
    stmt.close();
    conn.close();
}

public void add(String s1, String s2, String s3, String s4, Date s5, Boolean s6) throws SQLException {
    conn();
    stmt = conn.prepareStatement("INSERT INTO usersinformation (name,email,password,phone,birthday,male,balance)" + "values (?, ?, ?, ?, ?, ?, ?)");
    stmt.setString(1, s1);
    stmt.setString(2, s2);
    stmt.setString(3, s3);
    stmt.setString(4, s4);
    stmt.setDate(5, s5);
    stmt.setBoolean(6, s6);
    stmt.setDouble(7, 0);
    stmt.execute();
    conn.close();
}
```

Activate Wi
Go to Settings t