**Group members:**

Asna Aqeel (SP24-BSE-016)

Muhibba Fatima (SP24-BSE-138)

Rutaba Hayat (FA23-BSE-220)

**Project Tittle: “Online Wallet App”**

**Submitted To:**

Sir Shahid Bhatti

**Date: 19/12/2024**

**User Class:**

import java.io.Serializable;

public class User implements Serializable {

private String fullName;

private String username;

private String password;

private String email;

private String city;

private String gender;

private double balance;

private String pin;

public User(String username, String password, double balance, String pin, String fullName, String email, String city, String gender) {

this.username = username;

this.password = password;

this.balance = balance;

this.pin = pin;

this.fullName = fullName;

this.email = email;

this.gender = gender;

this.city = city;

}

public User(String username, double initialBalance){

this.username = username;

this.balance = initialBalance;

}

public String getPin() {

return pin;

}

public void setPin(String pin) {

this.pin = pin;

}

public String getPassword() {

return password;

}

public String getGender() {

return gender;

}

public double getBalance() {

return balance;

}

public void setBalance(double balance) {

this.balance = balance;

}

public void setPassword(String newPassword) {

this.password = newPassword;

}

}

**User Manager Class:**

import java.io.\*;

import java.util.HashMap;

public class UserManager {

private HashMap<String, User> users;

private final String dataFilePath = "file.dat";

public UserManager() {

users = readUserDataFromFile();

}

public boolean authenticateUser(String username, String password) {

User user = users.get(username);

return user != null && user.getPassword().equals(password);

}

public void createAccount(String username, String password, double balance, String pin, String fullName, String email, String city, String gender) {

if (users.containsKey(username)) {

throw new IllegalArgumentException("Username already exists.");

}

users.put(username, new User(username, password, balance, pin, fullName, email, city, gender));

saveUserDataToFile();

}

public boolean verifyMpin(String username, String enteredMpin) {

User user = users.get(username);

if (user != null) {

return user.getPin().equals(enteredMpin);

}

return false;

}

public boolean isValidPassword(String password) {

if (password.length() < 8) {

return false;

}

boolean hasLetter = false;

boolean hasDigit = false;

boolean hasSpecialChar = false;

for (char c : password.toCharArray()) {

if (Character.isLetter(c)) {

hasLetter = true;

} else if (Character.isDigit(c)) {

hasDigit = true;

} else if (!Character.isLetterOrDigit(c)) {

hasSpecialChar = true;

}

}

return hasLetter && hasDigit && hasSpecialChar;

}

public double getUserBalance(String username) {

User user = users.get(username);

return user.getBalance();

}

public boolean transferFunds(String senderUsername, String recipientUsername, double amount) {

User sender = users.get(senderUsername);

User recipient = users.get(recipientUsername);

if (sender == null || recipient == null) {

return false;

}

if (sender.getBalance() < amount) {

return false;

}

sender.setBalance(sender.getBalance() - amount);

recipient.setBalance(recipient.getBalance() + amount);

return true;

}

public User getUser(String username) {

return users.get(username);

}

public void deleteAccount(String username) {

users.remove(username);

saveUserDataToFile();

}

private HashMap<String, User> readUserDataFromFile() {

try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(dataFilePath))) {

return (HashMap<String, User>) ois.readObject();

} catch (IOException | ClassNotFoundException e) {

return new HashMap<>();

}

}

public void saveUserDataToFile() {

try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(dataFilePath))) {

oos.writeObject(users);

} catch (IOException e) {

throw new RuntimeException("Error saving user data to file", e);

}

}

}

**ATM APP Class:**

import javafx.application.Application;

import javafx.geometry.Insets;

import javafx.geometry.Pos;

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.image.Image;

import javafx.scene.image.ImageView;

import javafx.scene.layout.\*;

import javafx.scene.text.Font;

import javafx.stage.Stage;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

public class ATMApp extends Application {

private UserManager userManager;

private Stage primaryStage;

private BorderPane mainLayout;

private Label messageLabel;

private ImageView backgroundImage;

private String loggedInUser = null;

@Override

public void start(Stage primaryStage) {

this.primaryStage = primaryStage;

userManager = new UserManager();

mainLayout = new BorderPane();

showMainMenu(primaryStage);

mainLayout.setBottom(createToolBar());

}

private ToolBar createToolBar() {

Button homeButton = new Button();

homeButton.setStyle("-fx-background-color: transparent;");

Image homeIcon = new Image("file:/C:/Users/Admin/Downloads/homeIcon.png");

ImageView homeIconView = new ImageView(homeIcon);

homeIconView.setFitHeight(40);

homeIconView.setPreserveRatio(true);

homeButton.setGraphic(homeIconView);

homeButton.setOnAction(e -> showMainMenu(primaryStage));

Button feedbackButton = new Button();

feedbackButton.setStyle("-fx-background-color: transparent;");

Image feedbackIcon = new Image("file:/C:/Users/Admin/Downloads/feedback.png");

ImageView feedbackIconView = new ImageView(feedbackIcon);

feedbackIconView.setFitHeight(40);

feedbackIconView.setPreserveRatio(true);

feedbackButton.setGraphic(feedbackIconView);

feedbackButton.setOnAction(e ->{

if(loggedInUser != null) {

showFeedbackForm();

} else{

messageLabel.setText("Please login to provide feedback.");

messageLabel.setStyle("-fx-text-fill: #f44336; -fx-font-size: 24px; -fx-font-weight: bold;");

}

} );

Button faqButton = new Button();

faqButton.setStyle("-fx-background-color: transparent;");

Image faqIcon = new Image("file:/C:/Users/Admin/Downloads/faq.png");

ImageView faqIconView = new ImageView(faqIcon);

faqIconView.setFitHeight(40);

faqIconView.setPreserveRatio(true);

faqButton.setGraphic(faqIconView);

faqButton.setOnAction(e -> {

if (loggedInUser != null) {

showFAQs();

} else {

messageLabel.setText("Please login for frequently asked questions faq's section.");

messageLabel.setStyle("-fx-text-fill: #f44336; -fx-font-size: 24px; -fx-font-weight: bold;");

}

});

Button logoutButton = new Button();

logoutButton.setStyle("-fx-background-color: transparent;");

Image logoutIcon = new Image("file:/C:/Users/Admin/Downloads/logout.png");

ImageView logoutIconView = new ImageView(logoutIcon);

logoutIconView.setFitHeight(40);

logoutIconView.setPreserveRatio(true);

logoutButton.setGraphic(logoutIconView);

logoutButton.setOnAction(e -> System.exit(0));

ToolBar toolBar = new ToolBar(homeButton, feedbackButton, faqButton, logoutButton);

toolBar.setStyle("-fx-background-color: #a1d70a;");

toolBar.setPrefHeight(70);

return toolBar;

}

private void showMainMenu(Stage primaryStage) {

primaryStage.setTitle("Online Wallet");

VBox menuBox = new VBox(10);

menuBox.setAlignment(Pos.CENTER);

menuBox.setPadding(new Insets(20));

menuBox.setStyle("-fx-border-color: #a1d70a; -fx-background-color: rgba(0,0,0,0.6); -fx-border-width: 5; -fx-background-radius: 10;");

menuBox.setMinHeight(150);

menuBox.setMinWidth(250);

Button createAccountButton = new Button("Create an Account");

createAccountButton.setStyle("-fx-background-color: #a1d70a; -fx-text-fill: black; -fx-font-size: 18px; -fx-font-family: 'Times New Roman';");

createAccountButton.setPrefSize(200, 50);

Button loginButton = new Button("Login");

loginButton.setStyle("-fx-background-color: #a1d70a; -fx-text-fill: black; -fx-font-size: 18px; -fx-font-family: 'Times New Roman';");

loginButton.setPrefSize(200, 50);

menuBox.getChildren().addAll(createAccountButton, loginButton);

createAccountButton.setOnAction(e -> showCreateAccountPanel());

loginButton.setOnAction(e -> showLoginPanel());

if(backgroundImage == null) {

try {

Image image = new Image(new FileInputStream("C:\\Users\\Admin\\IdeaProjects\\demo32\\src\\main\\resources\\IMG\_5243.JPG"));

backgroundImage = new ImageView(image);

backgroundImage.setFitHeight(690);

backgroundImage.setFitWidth(1400);

} catch (FileNotFoundException e) {

throw new RuntimeException("Background image not found!", e);

}

}

if (mainLayout.getChildren().isEmpty()) {

StackPane root = new StackPane();

root.getChildren().addAll(backgroundImage, mainLayout);

Scene scene = new Scene(root, 1400, 690);

primaryStage.setScene(scene);

}

mainLayout.setCenter(menuBox);

messageLabel = new Label("Welcome to the Wallet");

messageLabel.setStyle("-fx-text-fill : #a1d70a; -fx-font-size: 44px; -fx-font-weight: bold; -fx-font-family: 'Times New Roman';");

mainLayout.setTop(messageLabel);

BorderPane.setAlignment(messageLabel, Pos.CENTER);

primaryStage.show();

}

private void showCreateAccountPanel() {

GridPane grid = createStyledGridPane();

Label fullNameLabel = createStyledLabel("Full name:", "#a1d70a");

fullNameLabel.setStyle("-fx-font-weight: bold; -fx-font-size: 20px; -fx-text-fill: #a1d70a;");

TextField fullNameField = new TextField();

Label usernameLabel = createStyledLabel("Username:", "#a1d70a");

usernameLabel.setStyle("-fx-font-weight: bold; -fx-font-size: 20px; -fx-text-fill: #a1d70a;");

TextField usernameField = new TextField();

Label passwordLabel = createStyledLabel("Password:", "#a1d70a");

passwordLabel.setStyle("-fx-font-weight: bold; -fx-font-size: 20px; -fx-text-fill: #a1d70a;");

PasswordField passwordField = new PasswordField();

Label pinLabel = createStyledLabel("PIN (4-digit):", "#a1d70a");

pinLabel.setStyle("-fx-font-weight: bold; -fx-font-size: 20px; -fx-text-fill: #a1d70a;");

PasswordField pinField = new PasswordField();

pinField.setPromptText("Enter 4-digit PIN");

Label emailLabel = createStyledLabel("Email", "#a1d70a");

emailLabel.setStyle("-fx-font-weight: bold; -fx-font-size: 20px; -fx-text-fill: #a1d70a;");

TextField emailField = new TextField();

Label balanceLabel = createStyledLabel("Initial Balance:", "#a1d70a");

balanceLabel.setStyle("-fx-font-weight: bold; -fx-font-size: 20px; -fx-text-fill: #a1d70a;");

TextField balanceField = new TextField();

Label genderLabel = new Label("Gender:");

genderLabel.setStyle("-fx-font-weight: bold; -fx-font-size: 20px; -fx-text-fill: #a1d70a;");

ToggleGroup genderGroup = new ToggleGroup();

RadioButton male = new RadioButton("Male");

RadioButton female = new RadioButton("Female");

male.setToggleGroup(genderGroup);

female.setToggleGroup(genderGroup);

male.setStyle("-fx-text-fill: a1d70a; -fx-font-size: 20px; -fx-font-weight: bold;");

female.setStyle("-fx-text-fill: a1d70a; -fx-font-size: 20px; -fx-font-weight: bold;");

Label cityLabel = new Label("City:");

cityLabel.setStyle("-fx-font-weight: bold; -fx-font-size: 20px; -fx-text-fill: #a1d70a;");

ComboBox<String> citybox = new ComboBox<>();

citybox.getItems().addAll("Gujranwala", "Lahore", "Islamabad", "Karachi", "Peshawar", "Quetta", "Sialkot");

Label accountTypeLabel = createStyledLabel("Account Type:", "#a1d70a");

accountTypeLabel.setStyle("-fx-font-weight: bold; -fx-font-size: 20px; -fx-text-fill: #a1d70a;");

ComboBox<String> accountTypeComboBox = new ComboBox<>();

accountTypeComboBox.getItems().addAll("Savings Account", "Current Account");

Button createButton = new Button("Create");

createButton.setStyle("-fx-background-color: #a1d70a; -fx-text-fill: black; -fx-font-weight: bold; -fx-font-size: 18px; -fx-font-family: 'Times New Roman';");

createButton.setPrefHeight(50);

createButton.setPrefWidth(200);

Button backButton = new Button("Back");

backButton.setStyle("-fx-background-color: #a1d70a; -fx-text-fill: black; -fx-font-weight: bold; -fx-font-size: 18px; -fx-font-family: 'Times New Roman';");

backButton.setPrefWidth(200);

backButton.setPrefHeight(50);

createButton.setOnAction(e -> {

try {

String fullName = fullNameField.getText();

String username = usernameField.getText();

String password = passwordField.getText();

String email = emailField.getText();

String city = citybox.getValue().toString();

String gender = male.isSelected() ? "Male" : "Female";

double balance = Double.parseDouble(balanceField.getText());

String pin = pinField.getText();

if(!userManager.isValidPassword(password)){

messageLabel.setText("Password must be at least 8 characters long and include letters, digits, and special characters.");

messageLabel.setStyle("-fx-text-fill: #f44336; -fx-font-sie: 24px; -fx-font-weight: bold;");

return;

}

if(!pin.matches("\\d{4}")){

messageLabel.setText("Error: PIN must be a 4 digit number.");

return;

}

userManager.createAccount(username, password, balance, pin, fullName, email, city, gender);

messageLabel.setText("Account created successfully.");

messageLabel.setStyle("-fx-text-fill: #f44336; -fx-font-weight: bold; -fx-font-size: 24px;");

showMainMenu(primaryStage);

} catch (Exception ex) {

messageLabel.setText("Error: " + ex.getMessage());

messageLabel.setStyle("-fx-text-fill: #f44336;");

}

});

backButton.setOnAction(e -> showMainMenu(primaryStage));

HBox genderBox = new HBox(10, male, female);

grid.add(fullNameLabel, 0,0);

grid.add(fullNameField, 1,0);

grid.add(usernameLabel, 0, 1);

grid.add(usernameField, 1, 1);

grid.add(passwordLabel, 0, 2);

grid.add(passwordField, 1, 2);

grid.add(emailLabel, 0,3);

grid.add(emailField, 1,3);

grid.add(pinLabel, 0,4);

grid.add(pinField, 1,4);

grid.add(balanceLabel, 0, 5);

grid.add(balanceField, 1, 5);

grid.add(genderLabel, 0, 6);

grid.add(genderBox, 1,6);

grid.add(cityLabel, 0, 7);

grid.add(citybox, 1, 7);

grid.add(accountTypeLabel, 0, 8);

grid.add(accountTypeComboBox, 1, 8);

grid.add(createButton, 0, 9);

grid.add(backButton, 1, 9);

mainLayout.setCenter(grid);

mainLayout.setBottom(createToolBar());

}

private void showLoginPanel() {

GridPane grid = createStyledGridPane();

Label usernameLabel = createStyledLabel("Username:", "#a1d70a");

usernameLabel.setStyle("-fx-font-weight: bold; -fx-font-size: 20px; -fx-text-fill: #a1d70a;");

TextField usernameField = new TextField();

Label passwordLabel = createStyledLabel("Password:", "#a1d70a");

passwordLabel.setStyle("-fx-font-weight: bold; -fx-font-size: 20px; -fx-text-fill: #a1d70a;");

PasswordField passwordField = new PasswordField();

Button loginButton = new Button("Login");

loginButton.setStyle("-fx-background-color: #a1d70a; -fx-text-fill: black; -fx-font-weight: bold; -fx-font-size: 18px; -fx-font-family: 'Times New Roman';");

loginButton.setPrefHeight(50);

loginButton.setPrefWidth(200);

Button backButton = new Button("Back");

backButton.setStyle("-fx-background-color: #a1d70a; -fx-text-fill: black; -fx-font-weight: bold; -fx-font-size: 18px; -fx-font-family: 'Times New Roman';");

backButton.setPrefWidth(200);

backButton.setPrefHeight(50);

loginButton.setOnAction(e -> {

try {

String username = usernameField.getText();

String password = passwordField.getText();

if (userManager.authenticateUser(username, password)) {

loggedInUser = username;

User user = userManager.getUser(username);

if (user != null) {

if ("Male".equals(user.getGender())) {

showMaleScreen();

} else if ("Female".equals(user.getGender())) {

showFemaleScreen();

} else {

messageLabel.setText("Error: Gender not set.");

messageLabel.setStyle("-fx-text-fill: #f44336;");

}

}

} else {

messageLabel.setText("Invalid username or password.");

messageLabel.setStyle("-fx-text-fill: #f44336; -fx-font-size: 40px; -fx-font-weight: bold");

}

} catch (Exception ex) {

messageLabel.setText("Error: " + ex.getMessage());

messageLabel.setStyle("-fx-text-fill: #f44336; -fx-font-size: 40px; -fx-font-weight: bold");

}

});

backButton.setOnAction(e -> showMainMenu(primaryStage));

grid.add(usernameLabel, 0, 0);

grid.add(usernameField, 1, 0);

grid.add(passwordLabel, 0, 1);

grid.add(passwordField, 1, 1);

grid.add(loginButton, 0, 2);

grid.add(backButton, 1, 2);

mainLayout.setCenter(grid);

mainLayout.setBottom(createToolBar());

}

private void showATMMenu(String username) {

GridPane gridPane = new GridPane();

gridPane.setPadding(new Insets(20));

gridPane.setHgap(20);

gridPane.setVgap(20);

gridPane.setAlignment(Pos.CENTER);

Button withdrawButton = new Button("Withdraw");

withdrawButton.setStyle("-fx-background-color: #a1d70a; -fx-text-fill: black; -fx-font-weight: bold; -fx-font-size: 18px; -fx-font-family: 'Times New Roman';");

withdrawButton.setPrefHeight(60);

withdrawButton.setPrefWidth(300);

Button depositButton = new Button("Deposit");

depositButton.setStyle("-fx-background-color: #a1d70a; -fx-text-fill: black; -fx-font-weight: bold; -fx-font-size: 18px; -fx-font-family: 'Times New Roman';");

depositButton.setPrefHeight(60);

depositButton.setPrefWidth(300);

Button balanceButton = new Button("Balance Inquiry");

balanceButton.setStyle("-fx-background-color: #a1d70a; -fx-text-fill: black; -fx-font-weight: bold; -fx-font-size: 18px; -fx-font-family: 'Times New Roman';");

balanceButton.setPrefWidth(300);

balanceButton.setPrefHeight(60);

Button transferButton = new Button("Transfer Funds");

transferButton.setStyle("-fx-background-color: #a1d70a; -fx-text-fill: black; -fx-font-weight: bold; -fx-font-size: 18px; -fx-font-family: 'Times New Roman';");

transferButton.setPrefHeight(60);

transferButton.setPrefWidth(300);

Button deleteButton = new Button("Delete Account");

deleteButton.setStyle("-fx-background-color: #a1d70a; -fx-text-fill: black; -fx-font-weight: bold; -fx-font-size: 18px; -fx-font-family: 'Times New Roman';");

deleteButton.setPrefWidth(300);

deleteButton.setPrefHeight(60);

Button changePassword = new Button("Change Password");

changePassword.setStyle("-fx-background-color: #a1d70a; -fx-text-fill: black; -fx-font-weight: bold; -fx-font-size: 18px; -fx-font-family: 'Times New Roman';");

changePassword.setPrefHeight(60);

changePassword.setPrefWidth(300);

Button logoutButton = new Button("Logout");

logoutButton.setStyle("-fx-background-color: #a1d70a; -fx-text-fill: black; -fx-font-weight: bold; -fx-font-size: 18px; -fx-font-family: 'Times New Roman';");

logoutButton.setPrefWidth(300);

logoutButton.setPrefHeight(60);

Button emergencyButton = new Button("Loan");

emergencyButton.setStyle("-fx-background-color: #a1d70a; -fx-text-fill: black; -fx-font-weight: bold; -fx-font-size: 18px; -fx-font-family: 'Times New Roman';");

emergencyButton.setPrefHeight(60);

emergencyButton.setPrefWidth(300);

withdrawButton.setOnAction(e -> performWithdrawal(username));

depositButton.setOnAction(e -> performDeposit(username));

balanceButton.setOnAction(e -> checkBalance(username));

deleteButton.setOnAction(e -> deleteAccount(username));

changePassword.setOnAction(e -> changePassword(username));

transferButton.setOnAction(e -> performTransferFunds(username));

logoutButton.setOnAction(e -> showMainMenu(primaryStage));

emergencyButton.setOnAction(e -> emergencyWithdraw());

gridPane.add(withdrawButton, 0, 0);

gridPane.add(depositButton, 1, 0);

gridPane.add(balanceButton, 2, 0);

gridPane.add(transferButton, 0, 1);

gridPane.add(deleteButton, 1, 1);

gridPane.add(changePassword, 2, 1);

gridPane.add(emergencyButton, 0, 2);

gridPane.add(logoutButton, 1, 2);

mainLayout.setCenter(gridPane);

mainLayout.setBottom(createToolBar());

}

private GridPane createStyledGridPane() {

GridPane grid = new GridPane();

grid.setPadding(new Insets(20));

grid.setHgap(15);

grid.setVgap(15);

grid.setAlignment(Pos.CENTER);

return grid;

}

private Label createStyledLabel(String text, String color) {

Label label = new Label(text);

label.setFont(Font.font("Arial", 18));

label.setStyle("-fx-text-fill: " + color + ";");

return label;

}

private void performWithdrawal(String username) {

try {

TextInputDialog dialog = new TextInputDialog();

dialog.setTitle("Withdrawal");

dialog.setHeaderText("Enter withdrawal amount:");

dialog.setContentText("Amount:");

dialog.showAndWait().ifPresent(amountStr -> {

try {

double amount = Double.parseDouble(amountStr);

User user = userManager.getUser(username);

if (amount <= 0 || amount > user.getBalance()) {

messageLabel.setText("Error: Invalid withdrawal amount.");

messageLabel.setStyle("-fx-text-fill: #f44336; -fx-font-size: 40px; -fx-font-weight: bold");

return;

}

user.setBalance(user.getBalance() - amount);

userManager.saveUserDataToFile();

messageLabel.setText("Withdrawal successful. Remaining balance: " + user.getBalance());

messageLabel.setStyle("-fx-text-fill: #a1d70a; -fx-font-size: 40px; -fx-font-weight: bold");

} catch (NumberFormatException ex) {

messageLabel.setText("Error: Invalid input. Please enter a number.");

messageLabel.setStyle("-fx-text-fill: #f44336;");

}

});

} catch (Exception e) {

messageLabel.setText("Error: " + e.getMessage());

messageLabel.setStyle("-fx-text-fill: #f44336;");

}

}

private void performDeposit(String username) {

try {

TextInputDialog dialog = new TextInputDialog();

dialog.setTitle("Deposit");

dialog.setHeaderText("Enter deposit amount:");

dialog.setContentText("Amount:");

dialog.showAndWait().ifPresent(amountStr -> {

try {

double amount = Double.parseDouble(amountStr);

if (amount <= 0) {

messageLabel.setText("Error: Deposit amount must be positive.");

messageLabel.setStyle("-fx-text-fill: #f44336; -fx-font-size: 40px; -fx-font-weight: bold");

return;

}

User user = userManager.getUser(username);

user.setBalance(user.getBalance() + amount);

userManager.saveUserDataToFile();

messageLabel.setText("Deposit successful. New balance: " + user.getBalance());

messageLabel.setStyle("-fx-text-fill: #a1d70a; -fx-font-size: 40px; -fx-font-weight: bold");

} catch (NumberFormatException ex) {

messageLabel.setText("Error: Invalid input. Please enter a number.");

messageLabel.setStyle("-fx-text-fill: #f44336; -fx-font-size: 40px; -fx-font-weight: bold");

}

});

} catch (Exception e) {

messageLabel.setText("Error: " + e.getMessage());

messageLabel.setStyle("-fx-text-fill: #f44336;");

}

}

private void checkBalance(String username) {

try {

User user = userManager.getUser(username);

messageLabel.setText("Current balance: " + user.getBalance());

messageLabel.setStyle("-fx-text-fill: #a1d70a; -fx-font-size: 40px; -fx-font-weight: bold");

} catch (Exception e) {

messageLabel.setText("Error: " + e.getMessage());

messageLabel.setStyle("-fx-text-fill: #f44336; -fx-font-size: 40px; -fx-font-weight: bold");

}

}

private void deleteAccount(String username) {

VBox confirmationBox = new VBox(20);

confirmationBox.setAlignment(Pos.CENTER);

confirmationBox.setPadding(new Insets(20));

Label confirmationLabel = new Label("Are you sure you want to delete your account?");

confirmationLabel.setStyle("-fx-font-size: 18px; -fx-text-fill: #f44336; -fx-font-weight: bold;");

Label warningLabel = new Label("This action cannot be undone.");

warningLabel.setStyle("-fx-font-size: 14px; -fx-text-fill: #f44336;");

Button confirmButton = new Button("Confirm");

confirmButton.setStyle("-fx-background-color: a1d70a; -fx-text-fill: #ff4d4d; -fx-font-weight: bold; -fx-font-size: 18px; -fx-font-family: 'Times New Roman'; ");

confirmButton.setPrefHeight(50);

confirmButton.setPrefWidth(200);

Button cancelButton = new Button("Cancel");

cancelButton.setStyle("-fx-background-color: a1d70a; -fx-text-fill: #ff4d4d; -fx-font-weight: bold; -fx-font-size: 18px; -fx-font-family: 'Times New Roman';");

cancelButton.setPrefWidth(200);

cancelButton.setPrefHeight(50);

confirmButton.setOnAction(e -> {

try {

userManager.deleteAccount(username);

messageLabel.setText("Account deleted successfully.");

messageLabel.setStyle("-fx-text-fill: #a1d70a;");

showMainMenu(primaryStage);

} catch (Exception ex) {

messageLabel.setText("Error: " + ex.getMessage());

messageLabel.setStyle("-fx-text-fill: #f44336;");

}

});

cancelButton.setOnAction(e -> showATMMenu(username));

confirmationBox.getChildren().addAll(confirmationLabel, warningLabel, confirmButton, cancelButton);

mainLayout.setCenter(confirmationBox);

}

private void changePassword(String username) {

try {

TextInputDialog currentPasswordDialog = new TextInputDialog();

currentPasswordDialog.setTitle("Change Password");

currentPasswordDialog.setHeaderText("Enter current password:");

currentPasswordDialog.setContentText("Current Password:");

currentPasswordDialog.showAndWait().ifPresent(currentPassword -> {

try {

User user = userManager.getUser(username);

if (!user.getPassword().equals(currentPassword)) {

messageLabel.setText("Error: Current password is incorrect.");

messageLabel.setStyle("-fx-text-fill: #f44336; -fx-font-size: 40px; -fx-font-weight: bold");

return;

}

TextInputDialog newPasswordDialog = new TextInputDialog();

newPasswordDialog.setTitle("Change Password");

newPasswordDialog.setHeaderText("Enter new password:");

newPasswordDialog.setContentText("New Password:");

newPasswordDialog.showAndWait().ifPresent(newPassword -> {

try {

if (newPassword.isEmpty()) {

messageLabel.setText("Error: Password cannot be empty.");

messageLabel.setStyle("-fx-text-fill: #f44336; -fx-font-size: 40px; -fx-font-weight: bold");

return;

}

user.setPassword(newPassword);

userManager.saveUserDataToFile();

messageLabel.setText("Password changed successfully.");

messageLabel.setStyle("-fx-text-fill: #a1d70a; -fx-font-size: 40px; -fx-font-weight: bold");

} catch (Exception ex) {

messageLabel.setText("Error: " + ex.getMessage());

messageLabel.setStyle("-fx-text-fill: #f44336; -fx-font-size: 40px; -fx-font-weight: bold");

}

});

} catch (Exception ex) {

messageLabel.setText("Error: " + ex.getMessage());

messageLabel.setStyle("-fx-text-fill: #f44336; -fx-font-size: 40px; -fx-font-weight: bold");

}

});

} catch (Exception e) {

messageLabel.setText("Error: " + e.getMessage());

messageLabel.setStyle("-fx-text-fill: #f44336;");

}

}

private void showFeedbackForm() {

VBox feedbackLayout = new VBox(20);

feedbackLayout.setAlignment(Pos.CENTER);

feedbackLayout.setPadding(new Insets(20));

Label feedbackLabel = new Label("We value your feedback!");

feedbackLabel.setStyle("-fx-font-size: 24px; -fx-font-weight: bold; -fx-text-fill: #a1d70a;");

TextArea feedbackTextArea = new TextArea();

feedbackTextArea.setPromptText("Enter your feedback here...");

feedbackTextArea.setWrapText(true);

feedbackTextArea.setPrefHeight(150);

feedbackTextArea.setStyle("-fx-font-size: 16px;");

Button submitButton = new Button("Submit Feedback");

submitButton.setStyle("-fx-background-color: #a1d70a; -fx-text-fill: black; -fx-font-weight: bold; -fx-font-size: 22px; -fx-font-family: 'Times New Roman';");

submitButton.setPrefWidth(200);

submitButton.setOnAction(e -> {

String feedback = feedbackTextArea.getText();

if (feedback.isEmpty()) {

messageLabel.setText("Please provide some feedback.");

messageLabel.setStyle("-fx-text-fill: #f44336;");

} else {

messageLabel.setText("Thank you for your feedback!");

messageLabel.setStyle("-fx-text-fill: #a1d70a;");

}

});

Button backButton = new Button("Back");

backButton.setStyle("-fx-background-color: #a1d70a; -fx-text-fill: black; -fx-font-weight: bold; -fx-font-size: 18px; -fx-font-family: 'Times New Roman';");

backButton.setPrefWidth(200);

backButton.setOnAction(e -> showMainMenu(primaryStage));

feedbackLayout.getChildren().addAll(feedbackLabel, feedbackTextArea, submitButton, backButton);

mainLayout.setCenter(feedbackLayout);

}

private void showWelcomePage(Stage primaryStage, String username) {

VBox vBox = new VBox(20);

vBox.setAlignment(Pos.CENTER);

Label welcomeLabel = new Label("Welcome " + username + " to the Wallet.");

welcomeLabel.setStyle("-fx-text-fill: #a1d70a; -fx-font-size: 24px; -fx-font-family: 'Times New Roman'; -fx-font-weight: bold;");

Button okButton = new Button("Ok");

okButton.setStyle("-fx-background-color: #a1d70a; -fx-text-fill: black; -fx-font-size: 18px; -fx-font-family: 'Times New Roman'; -fx-font-weight: bold;");

okButton.setOnAction(e ->{

showATMMenu(username);

primaryStage.close();

});

vBox.getChildren().addAll(welcomeLabel, okButton);

vBox.setStyle("-fx-background-color: #1c1c1c; -fx-border-width: 3px; -fx-border-color: #a1d70a;");

Scene scene = new Scene(vBox, 400, 300);

primaryStage.setScene(scene);

primaryStage.show();

}

private void showMpinScreen(String username) {

GridPane grid = createStyledGridPane();

Label mpinLabel = createStyledLabel("Enter 4-digit MPIN:", "#a1d70a");

PasswordField mpinField = new PasswordField();

mpinField.setPromptText("4-digit MPIN");

Button submitButton = new Button("Submit");

submitButton.setStyle("-fx-background-color: #a1d70a; -fx-text-fill: black; -fx-font-weight: bold; -fx-font-size: 18px; -fx-font-family: 'Times New Roman';");

submitButton.setPrefHeight(50);

submitButton.setPrefWidth(200);

Button backButton = new Button("Back");

backButton.setStyle("-fx-background-color: #a1d70a; -fx-text-fill: black; -fx-font-weight: bold; -fx-font-size: 18px; -fx-font-family: 'Times New Roman';");

backButton.setPrefHeight(50);

backButton.setPrefWidth(200);

submitButton.setOnAction(e -> {

String enteredMpin = mpinField.getText();

if (userManager.verifyMpin(username, enteredMpin)) {

messageLabel.setText("Welcome, " + username + "!");

messageLabel.setStyle("-fx-text-fill: #a1d70a; -fx-font-size: 40px; -fx-font-weight: bold;");

Stage newStage = new Stage();

showWelcomePage(newStage, username);

} else {

messageLabel.setText("Invalid MPIN. Please try again.");

messageLabel.setStyle("-fx-text-fill: #f44336; -fx-font-size: 22px; -fx-font-weight: bold;");

}

});

backButton.setOnAction(e -> showLoginPanel());

grid.add(mpinLabel, 0, 0);

grid.add(mpinField, 1, 0);

grid.add(submitButton, 0, 1);

grid.add(backButton, 1, 1);

mainLayout.setCenter(grid);

mainLayout.setBottom(createToolBar());

}

private void performTransferFunds(String senderUsername) {

GridPane grid = createStyledGridPane();

Label recipientLabel = createStyledLabel("Recipient Username:", "#a1d70a");

TextField recipientField = new TextField();

Label amountLabel = createStyledLabel("Amount to Transfer:", "#a1d70a");

TextField amountField = new TextField();

Button transferButton = new Button("Transfer");

transferButton.setStyle("-fx-background-color: #a1d70a; -fx-text-fill: black; -fx-font-weight: bold; -fx-font-size: 18px; -fx-font-family: 'Times New Roman';");

transferButton.setPrefHeight(50);

transferButton.setPrefWidth(200);

Button backButton = new Button("Back");

backButton.setStyle("-fx-background-color: #a1d70a; -fx-text-fill: black; -fx-font-weight: bold; -fx-font-size: 18px; -fx-font-family: 'Times New Roman';");

backButton.setPrefHeight(50);

backButton.setPrefWidth(200);

transferButton.setOnAction(e -> {

try {

String recipientUsername = recipientField.getText();

double amount = Double.parseDouble(amountField.getText());

if (recipientUsername.isEmpty()) {

messageLabel.setText("Recipient username cannot be empty.");

messageLabel.setStyle("-fx-text-fill: #f44336; -fx-font-weight: bold; -fx-font-sie: 24px;");

return;

}

if (amount <= 0) {

messageLabel.setText("Transfer amount must be greater than zero.");

messageLabel.setStyle("-fx-text-fill: #f44336; -fx-font-weight: bold; -fx-font-sie: 24px;");

return;

}

if (userManager.transferFunds(senderUsername, recipientUsername, amount)) {

messageLabel.setText("Funds transferred successfully!");

messageLabel.setStyle("-fx-text-fill: #a1d70a; -fx-font-weight: bold; -fx-font-sie: 24px;");

showATMMenu(senderUsername);

} else {

messageLabel.setText("Transfer failed. Check recipient and balance.");

messageLabel.setStyle("-fx-text-fill: #f44336; -fx-font-weight: bold; -fx-font-sie: 24px;");

}

} catch (NumberFormatException ex) {

messageLabel.setText("Invalid amount. Please enter a numeric value.");

messageLabel.setStyle("-fx-text-fill: #f44336; -fx-font-weight: bold; -fx-font-sie: 24px;");

} catch (Exception ex) {

messageLabel.setText("Error: " + ex.getMessage());

messageLabel.setStyle("-fx-text-fill: #f44336;");

}

});

backButton.setOnAction(e -> showATMMenu(senderUsername));

grid.add(recipientLabel, 0, 0);

grid.add(recipientField, 1, 0);

grid.add(amountLabel, 0, 1);

grid.add(amountField, 1, 1);

grid.add(transferButton, 0, 2);

grid.add(backButton, 1, 2);

mainLayout.setCenter(grid);

mainLayout.setBottom(createToolBar());

}

private void showFAQs() {

BorderPane faqLayout = new BorderPane();

faqLayout.setPadding(new Insets(20));

Label faqLabel = new Label("Frequently Asked Questions");

faqLabel.setStyle("-fx-font-size: 24px; -fx-font-weight: bold; -fx-text-fill: #a1d70a;");

BorderPane.setAlignment(faqLabel, Pos.CENTER);

faqLayout.setTop(faqLabel);

VBox faqContent = new VBox(15);

faqContent.setPadding(new Insets(10));

faqContent.setAlignment(Pos.TOP\_LEFT);

Label question1 = new Label("Q: How do I add funds to my wallet?");

Label answer1 = new Label("A: Go to the 'Add Funds' section, enter the amount, and select your payment method.");

Label question2 = new Label("Q: What happens if I forget my password?");

Label answer2 = new Label("A: Click on 'Forgot Password' on the login screen and follow the steps to reset it.");

Label question3 = new Label("Q: How can I contact support?");

Label answer3 = new Label("A: Use the 'Feedback' button to send us a message, or email support@yourapp.com.");

question1.setStyle("-fx-font-weight: bold; -fx-text-fill: #000000;");

answer1.setStyle("-fx-text-fill: #000000;");

question2.setStyle("-fx-font-weight: bold; -fx-text-fill: #000000;");

answer2.setStyle("-fx-text-fill: #000000;");

question3.setStyle("-fx-font-weight: bold; -fx-text-fill: #000000;");

answer3.setStyle("-fx-text-fill: #000000;");

faqContent.getChildren().addAll(question1, answer1, question2, answer2, question3, answer3);

ScrollPane scrollPane = new ScrollPane(faqContent);

scrollPane.setFitToWidth(true);

scrollPane.setStyle("-fx-background-color: transparent;");

faqLayout.setCenter(scrollPane);

Button backButton = new Button("Back");

backButton.setStyle("-fx-background-color: #a1d70a; -fx-text-fill: black; -fx-font-weight: bold; -fx-font-size: 18px; -fx-font-family: 'Times New Roman';");

backButton.setPrefWidth(200);

backButton.setOnAction(e -> showATMMenu(loggedInUser));

BorderPane.setAlignment(backButton, Pos.CENTER);

faqLayout.setBottom(backButton);

mainLayout.setCenter(faqLayout);

}

private void emergencyWithdraw(){

GridPane grid = createStyledGridPane();

Label titleLabel = createStyledLabel("Emergency Loan Request", "#a1d70a");

titleLabel.setStyle("-fx-text-fill: #a1d70a; -fx-font-size: 30px; -fx-font-weight: bold; -fx-font-family: 'Times New Roman';");

Label loanLimitLabel = new Label("Maximum Loan Limit: 5000");

loanLimitLabel.setStyle("-fx-font-weight: bold; -fx-font-size: 20px; -fx-text-fill: #a1d70a;");

Label loanAmountLabel = new Label("Enter Loan Amount:");

loanAmountLabel.setStyle("-fx-font-weight: bold; -fx-font-size: 20px; -fx-text-fill: #a1d70a");

TextField loanAmountField = new TextField();

loanAmountField.setPromptText("Enter amount...");

Button requestLoanButton = new Button("Request Loan");

requestLoanButton.setStyle("-fx-background-color: #a1d70a; -fx-text-fill: black; -fx-font-weight: bold; -fx-font-size: 18px; -fx-font-family: 'Times New Roman';");

requestLoanButton.setPrefHeight(50);

requestLoanButton.setPrefWidth(200);

Button backButton = new Button("Back");

backButton.setStyle("-fx-background-color: #a1d70a; -fx-text-fill: black; -fx-font-weight: bold; -fx-font-size: 18px; -fx-font-family: 'Times New Roman';");

backButton.setPrefHeight(50);

backButton.setPrefWidth(200);

Label messageLabel = new Label();

messageLabel.setStyle("-fx-text-fill: red; -fx-font-size: 18px; -fx-font-weight: bold;");

requestLoanButton.setOnAction(e -> {

try {

double loanAmount = Double.parseDouble(loanAmountField.getText());

double maxLoanLimit = 5000.0;

if (loanAmount <= 0) {

messageLabel.setText("Loan amount must be greater than zero.");

} else if (loanAmount > maxLoanLimit) {

messageLabel.setText("Requested amount exceeds the maximum loan limit.");

} else {

messageLabel.setStyle("-fx-text-fill: a1d70a; -fx-font-size: 18px; -fx-font-weight: bold;");

messageLabel.setText("Loan approved! Amount: PKR" + loanAmount);

processLoanForUser(loggedInUser, loanAmount);

}

} catch (NumberFormatException ex) {

messageLabel.setText("Please enter a valid numeric amount.");

}

});

backButton.setOnAction(e -> showATMMenu(loggedInUser));

grid.add(titleLabel, 0, 0, 2, 1);

grid.add(loanLimitLabel, 0, 1, 2, 1);

grid.add(loanAmountLabel, 0, 2);

grid.add(loanAmountField, 1, 2);

grid.add(requestLoanButton, 0, 3);

grid.add(backButton, 1, 3);

grid.add(messageLabel, 0, 4, 2, 1);

mainLayout.setCenter(grid);

}

private void processLoanForUser(String username, double loanAmount) {

User user = userManager.getUser(username);

if (user != null) {

double currentBalance = user.getBalance();

double maxLoanLimit = 5000.0;

if

(loanAmount > 0 && loanAmount <= maxLoanLimit) {

user.setBalance(currentBalance + loanAmount);

System.out.println("Loan of PKR" + loanAmount + " approved for " + username);

System.out.println("New Balance: PKR" + user.getBalance());

} else {

System.out.println("Loan amount exceeds the maximum limit of PKR" + maxLoanLimit);

}

} else {

System.out.println("User not found.");

}

}

private void showMaleScreen() {

VBox maleScreenLayout = new VBox(20);

maleScreenLayout.setAlignment(Pos.CENTER);

maleScreenLayout.setPadding(new Insets(20));

Image maleImage = new Image("file:/C:/Users/Admin/Downloads/male.png");

ImageView maleImageView = new ImageView(maleImage);

maleImageView.setFitWidth(200);

maleImageView.setFitHeight(200);

maleImageView.setPreserveRatio(true);

Label welcomeLabel = new Label("Hello, Mr.! " + loggedInUser);

welcomeLabel.setStyle("-fx-text-fill: #a1d70a; -fx-font-size: 24px; -fx-font-weight: bold;");

Button continueButton = new Button("Continue");

continueButton.setStyle("-fx-background-color: #a1d70a; -fx-text-fill: black; -fx-font-size: 18px;");

continueButton.setOnAction(e->{showMpinScreen(loggedInUser);});

maleScreenLayout.getChildren().addAll(maleImageView,welcomeLabel, continueButton);

mainLayout.setCenter(maleScreenLayout);

}

private void showFemaleScreen() {

VBox femaleScreenLayout = new VBox(20);

femaleScreenLayout.setAlignment(Pos.CENTER);

femaleScreenLayout.setPadding(new Insets(20));

Image femaleImage = new Image("file:/C:/Users/Admin/Downloads/femaleimgae.jpeg");

ImageView femaleImageView = new ImageView(femaleImage);

femaleImageView.setFitWidth(200);

femaleImageView.setFitHeight(200);

femaleImageView.setPreserveRatio(true);

Label welcomeLabel = new Label("Hello, Ms.! " + loggedInUser);

welcomeLabel.setStyle("-fx-text-fill: #a1d70a; -fx-font-size: 24px; -fx-font-weight: bold;");

Button continueButton = new Button("Continue");

continueButton.setStyle("-fx-background-color: #a1d70a; -fx-text-fill: black; -fx-font-size: 18px;");

continueButton.setOnAction(e -> showMpinScreen(loggedInUser));

femaleScreenLayout.getChildren().addAll(femaleImageView, welcomeLabel, continueButton);

mainLayout.setCenter(femaleScreenLayout);

}

public static void main(String[] args) {

launch(args);

}

}

**Output:**