

Project: Banking System

Sai Kiran Bandari, Sirisha G, Kalyan G, Tharun Bhashyam

Computer Information Systems & Information Technology

University of Central Missouri

CIS4610: Special Project

Professor. Javaria Ahmad

In financial institution the banking system is the main drive the institution. The system helps in providing very important tools for the individuals to manage their finances, facilitate transaction and access various banking services(Mehdiabadi, Tabatabeinasab, Spulbar & etal, 2020).. In the modern days sophisticated banking system have been developed to ensure security and easy functionality of the banking institution(Abdelsalam, Elnahass, Ahmed & etal 2022). Different banking systems are run on different platform including online banking system (Lopatin, 2019)..

The projects aim is to develop a banking system which feature that can enable customer transactions, account creating and reduced calculation within the banking the system.

Key Features:

Account Management: Allowing users to create, view, modify, and close their accounts online.

Providing features for balance inquiries, transaction histories, and statement generation.

Fund Transfer and Payments: Enabling users to transfer funds between accounts, make payments, and set up recurring transactions.

Technology Stack:

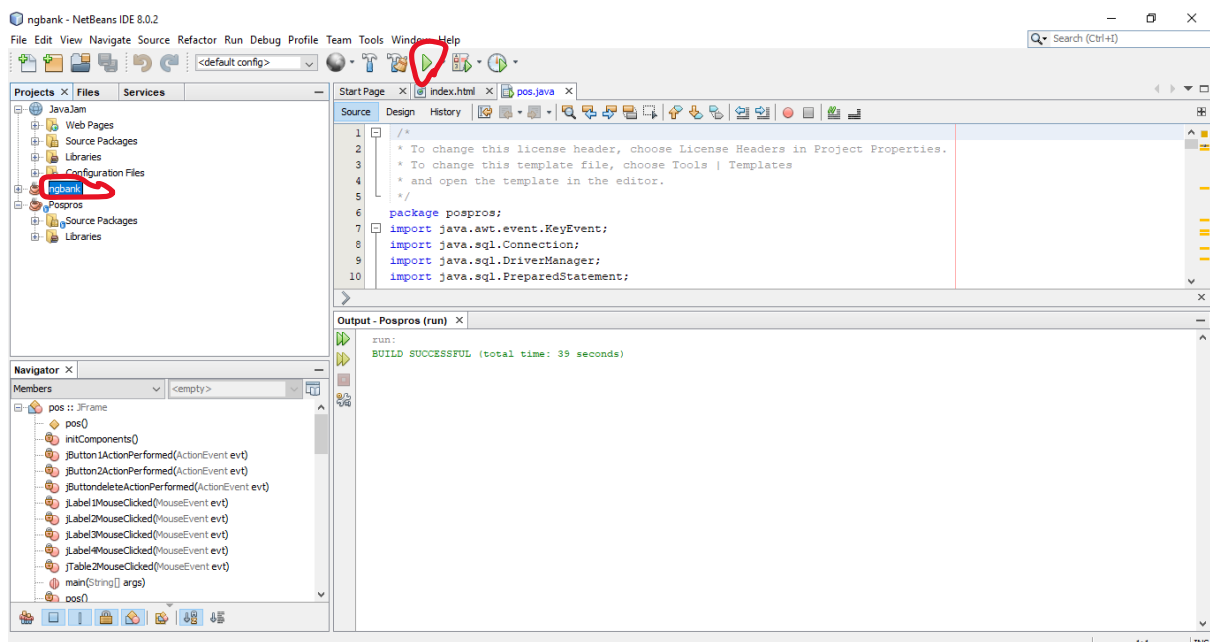
The banking system will be built using robust technologies such as:

Backend: Utilizing languages such as Java for the connection to the database (Tsindeliani, Proshunin, Sadovskaya & etal, 2022).

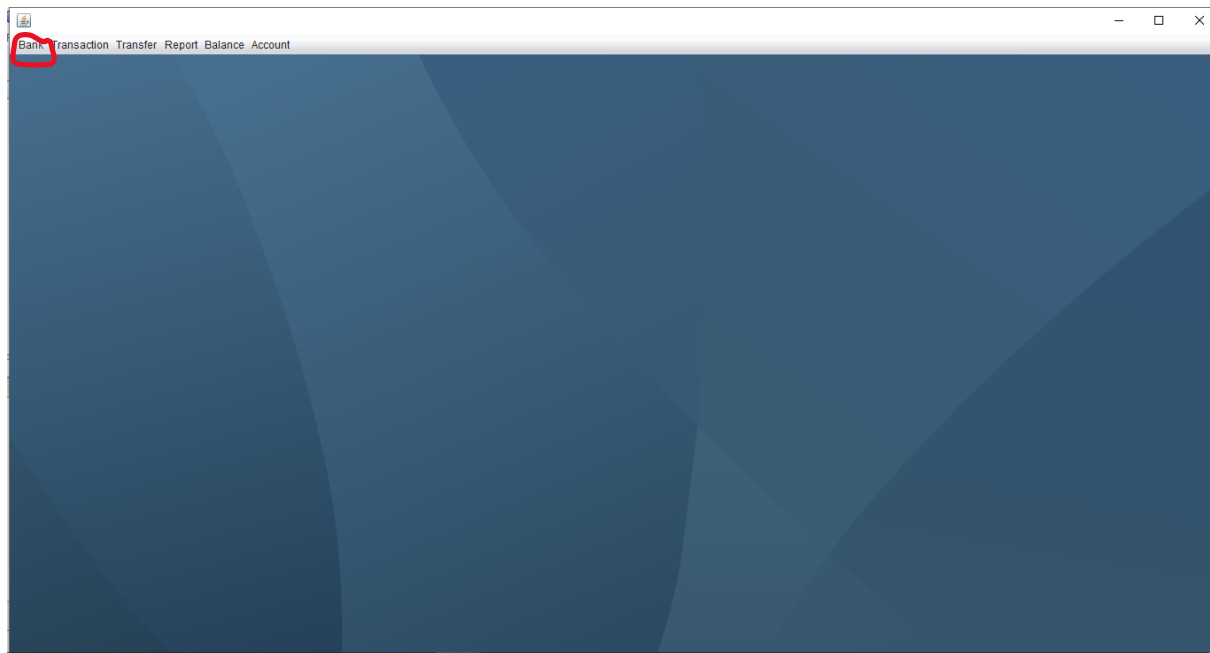
Database: Implementing secure database management systems like MySQL and SQL to store and manage user data.

Frontend: Utilizing java graphical user interface to develop the interface for the project. The graphical user interface for java famously known as GUI will be used in the development of the project's interfaces

The run and execution of the project starts by running from the NetBeans ide. By identifying the project to run from the NetBeans you click the project and then click run and build the project.



Once the system run it will show its first interface. From the interface you can now freely move within the banking system and making transaction creating a new account. below is the first interface of the banking system.



On clicking the bank heading two menu will appear one with customer and another with account heading respectively. if you click on the customer menu below is the interface of the system.

A screenshot of the 'Customer' registration interface. The window title bar shows the same menu as the first image. On the left is a dark teal sidebar with a dropdown arrow and four menu items: 'Register Customer', 'Print statement', 'Print ATM Card', and 'Exit'. The main area has a light gray header with the title 'Customer'. Below the header is a form titled 'Customer' with the following fields: 'Customer No' (containing 'CS0004'), 'First name', 'Second Name', 'Street', 'City', 'Branch' (a dropdown menu with 'emali' selected), and 'phone'. To the right of these fields are two image upload boxes labeled 'passport' and 'right thumb'. To the right of the 'passport' box is an 'Add' button. To the right of the 'right thumb' box is a 'Cancel' button.

Under this form you can create a new customer by uploading details about the customer in the give field. On click the account menu the account registration interface appears. In this is where

you create a new account number for the customer. The customer id is the unique id created in the customer interface.

The screenshot shows a web application window with a menu bar at the top containing 'Bank', 'Transaction', 'Transfer', 'Report', 'Balance', and 'Account'. A dark teal sidebar on the left contains the following menu items: 'Register Customer', 'Check Balance', 'Print statement', 'Print ATM Card', and 'Exit'. The main content area is titled 'Account' and contains a form with the following fields and controls:

- Account no:** A text input field containing 'AC00005'.
- Customer ID:** An empty text input field.
- Customer Name:** An empty text input field.
- Account type:** A dropdown menu with 'savings' selected.
- Balance:** An empty text input field.
- passport:** A large empty rectangular box.
- Buttons:** 'Find', 'Add', and 'Cancel' buttons are located on the right side of the form.

Under the transaction menu you can do deposits to customers account, withdraws from customer accounts. Depending on the account you pass to the account field. Below is the interface once you click deposit from the transaction menu

The screenshot shows a web application window with a menu bar at the top containing 'Bank', 'Transaction', 'Transfer', 'Report', 'Balance', and 'Account'. The main content area is titled 'Deposit' and contains the following sections:

- deposit:** A section containing a form with the title 'Enter Account number below', a text input field, and a 'Find' button.
- Balance:** A section containing a text input field labeled 'Balance'.
- account details:** A section containing a form with the following fields: 'Customer No' (with 'Customer Name' entered), 'First Name' (with 'First name' entered), 'Last Name' (with 'Last Name' entered), and 'Date' (with '2023-11-26' entered).
- Deposit:** A section containing a large teal rectangular button and two buttons labeled 'Deposit' and 'Cancel'.

On click the withdraw menu in transaction menu bar here its interface.

Bank Transaction Transfer Report Balance Account

Withdraw

withdraw

Enter Account number below

Find

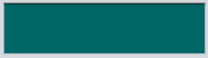
Balance

Balance

account details

Customer No	Customer Name
First Name	First name
Last Name	Last Name
Date	2023-11-26

Withdraw



Withdraw Cancel

Under transfer menu in the menu bar you can transfer funds from one account to another . you just need to enter the account to transfer from form and the account to transfer to.

Bank Transaction Transfer Report Balance Account

Transfer

Transfer

From account

Find

Balance

To account

Amount

Transfer

Cancel

Code structure and features.

The code is organized into classes where by each class has its own method and functionalities. The class includes customer, account, deposits, withdraw, transfer. Below is the code structure of the customer.

```

20  /**
21  */
22  public class customer extends javax.swing.JInternalFrame {
23
24      /**
25       * Creates new form customer
26       */
27      public customer() {
28          initComponents();
29          aoutID();
30          branch();
31      }
32      Connection con1;
33      PreparedStatement pst;
34      /**
35       * This method is called from within the constructor to initialize the form.
36       * WARNING: Do NOT modify this code. The content of this method is always
37       * regenerated by the Form Editor.
38       */
39      @SuppressWarnings("unchecked")
40
41
42      public void aoutID() {
43          try {
44              Class.forName("com.mysql.jdbc.Driver");
45              con1=DriverManager.getConnection("jdbc:mysql://localhost/ngbank", "root","");
46              Statement s = con1.createStatement();

```

The account class is organized as shown below.

```

22  public class account extends javax.swing.JInternalFrame {
23
24      /**
25       * Creates new form customer
26       */
27      public account() {
28          initComponents();
29          aoutID();
30          branch();
31      }
32      Connection con1;
33      PreparedStatement pst;
34      ResultSet rs;
35      /**
36       * This method is called from within the constructor to initialize the form.
37       * WARNING: Do NOT modify this code. The content of this method is always
38       * regenerated by the Form Editor.
39       */
40      @SuppressWarnings("unchecked")
41
42
43      public void aoutID() {
44          try {
45              Class.forName("com.mysql.jdbc.Driver");
46              con1=DriverManager.getConnection("jdbc:mysql://localhost/ngbank", "root","");
47              Statement s = con1.createStatement();
48
49              ResultSet rs = s.executeQuery("select Max(acc id) from account");

```

Steps in developing the code.

The coding started with understanding of the problem the project was working. The flow of data was designed and different class are formulated. Once the problem is understood the pseudocode is developed to give a blueprint of the system that's being developed. The scope of the project is declared. The project is reduced into manageable module that is class which

are easier to develop. Each class was tested on completion to ensure that is working as per the expectations.

References

- Abdelsalam, O., Elnahass, M., Ahmed, H., & Williams, J. (2022). Asset securitizations and bank stability: evidence from different banking systems. *Global finance journal*, 51, 100551.
- Lopatin, E. (2019). Assessment of Russian banking system performance and sustainability. *Banks and Bank Systems*, 14(3), 202.
- Mehdiabadi, A., Tabatabeinasab, M., Spulbar, C., Karbassi Yazdi, A., & Birau, R. (2020). Are we ready for the challenge of Banks 4.0? Designing a roadmap for banking systems in Industry 4.0. *International Journal of Financial Studies*, 8(2), 32.
- Tsindeliani, I. A., Proshunin, M. M., Sadovskaya, T. D., Popkova, Z. G., Davydova, M. A., & Babayan, O. A. (2022). Digital transformation of the banking system in the context of sustainable development. *Journal of Money Laundering Control*, 25(1), 165-180.