BANKING APPLICATION USING JDBC

PRESENTATION FLOW

→

1 Introduction

2 Project Objective

3 Features of the Banking Application

4 Technologies Used

5 System Architecture

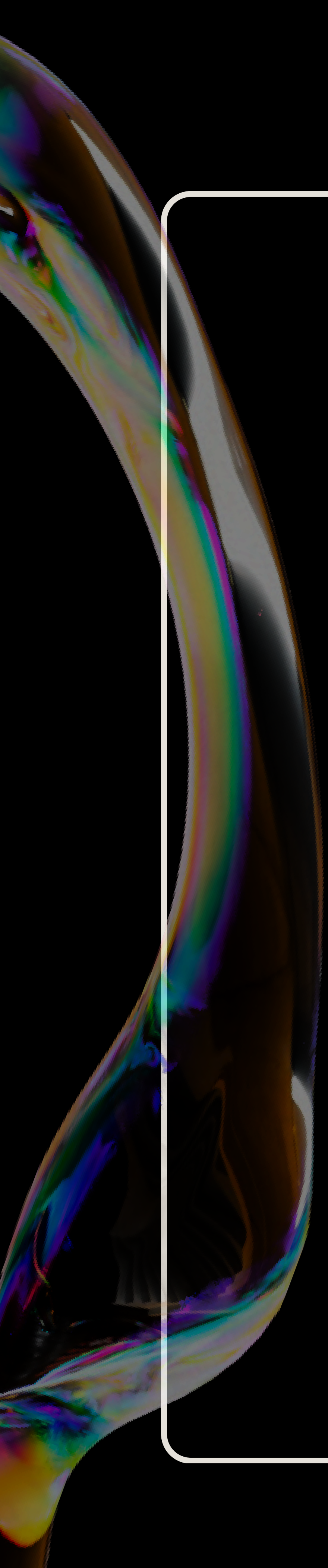
6 Database Schema

7 Flow of the Application

8 Challenges Faced

9 Conclusion

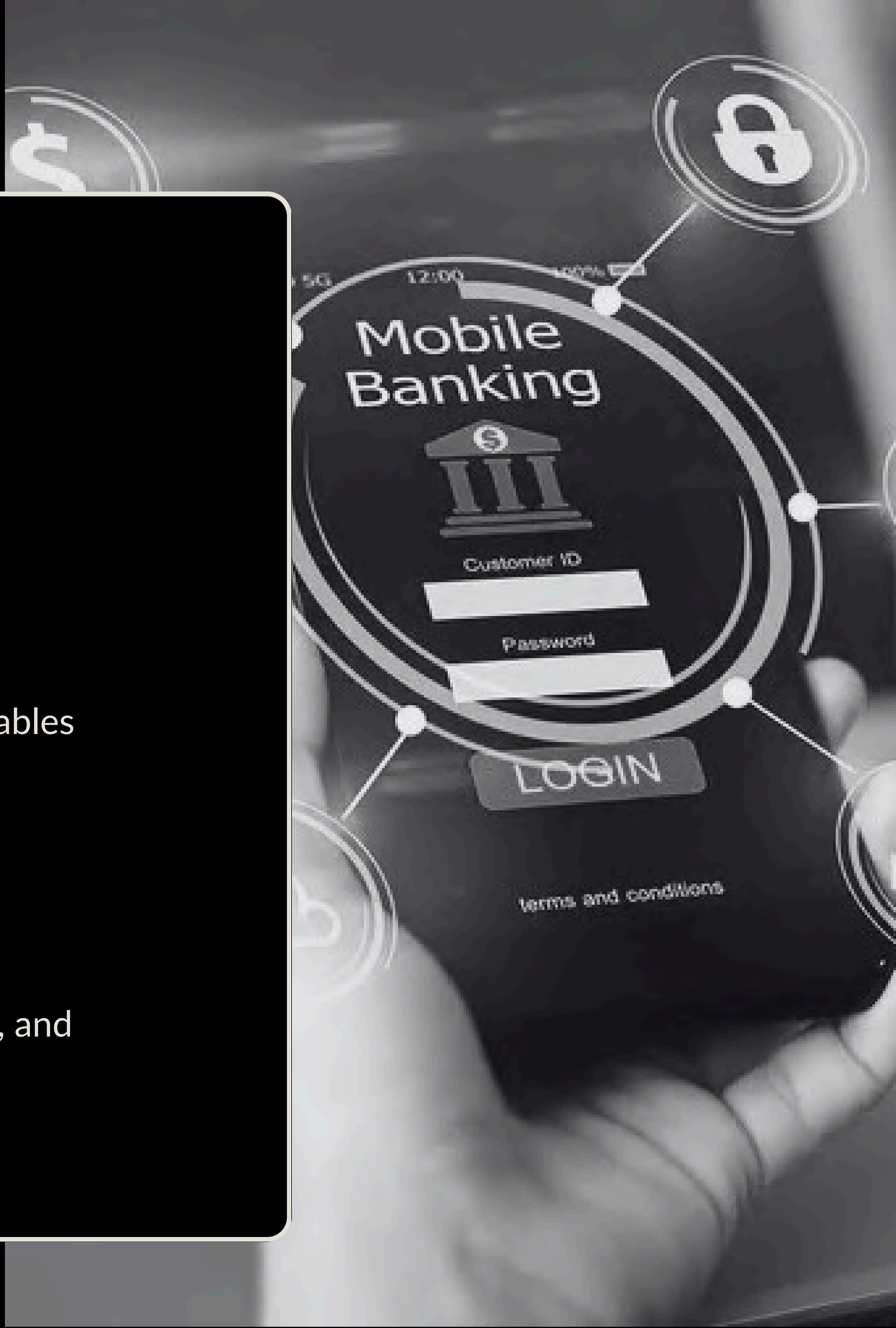
10 Team Contributions



INTRODUCTION

What is the Project?

- This project aims to develop a Banking Application that enables users to securely manage their accounts.
- Users can perform banking tasks like checking balances, transferring money, and viewing transaction histories.
- The system is designed with a focus on security, efficiency, and user experience.



PROJECT OBJECTIVE

Develop a robust Banking System with essential features

- Create Account: Users can create an account with a unique 3-digit account number.
- Login & Authentication: Secure login using the account number and password.
- Check Balance: View current balance in real-time.
- Transfer Money: Send money between accounts securely.
- Transaction History: View all past transactions with timestamps.

FEATURES OF THE BANKING APPLICATION

Features of the Banking Application

- Account Creation: Users can open an account by entering details and receiving a 3-digit unique account number.
- Secure Login: Login with password protection.
- Balance Inquiry: Users can check their current account balance anytime.
- Money Transfer: Secure fund transfers between users.
- Transaction History: Users can view all previous transactions with details like amount, sender, recipient, and date.

TECHNOLOGIES USED

JAVA

Java is a high-level, object-oriented programming language that's widely used for developing a variety of applications like this one.

MySQL

MySQL is a popular open-source relational database management system (RDBMS) used for storing and organizing data. We use MYSQL to store data.

JDBC

JDBC (Java Database Connectivity) is an API (Application Programming Interface) that provides a standard way for Java programs to interact with databases. It acts as a bridge between the Java application and the database system

SYSTEM ARCHITECTURE

Front-End

Java console-based interface.

Back-End

MySQL database storing user account details and transactions.

JDBC

Java Database Connectivity used to interact with MySQL for CRUD operations (Create, Read, Update, Delete).

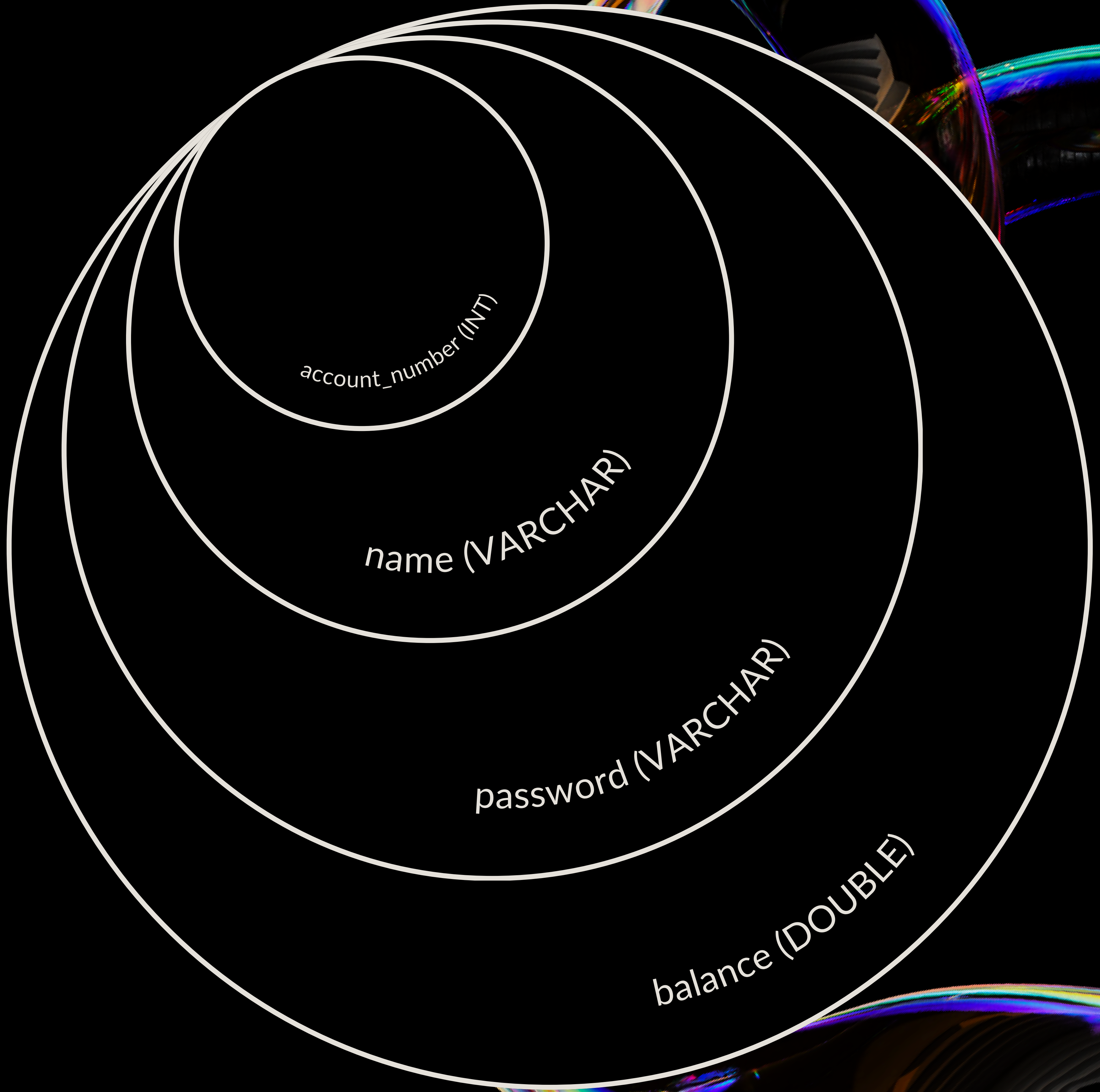
Framework

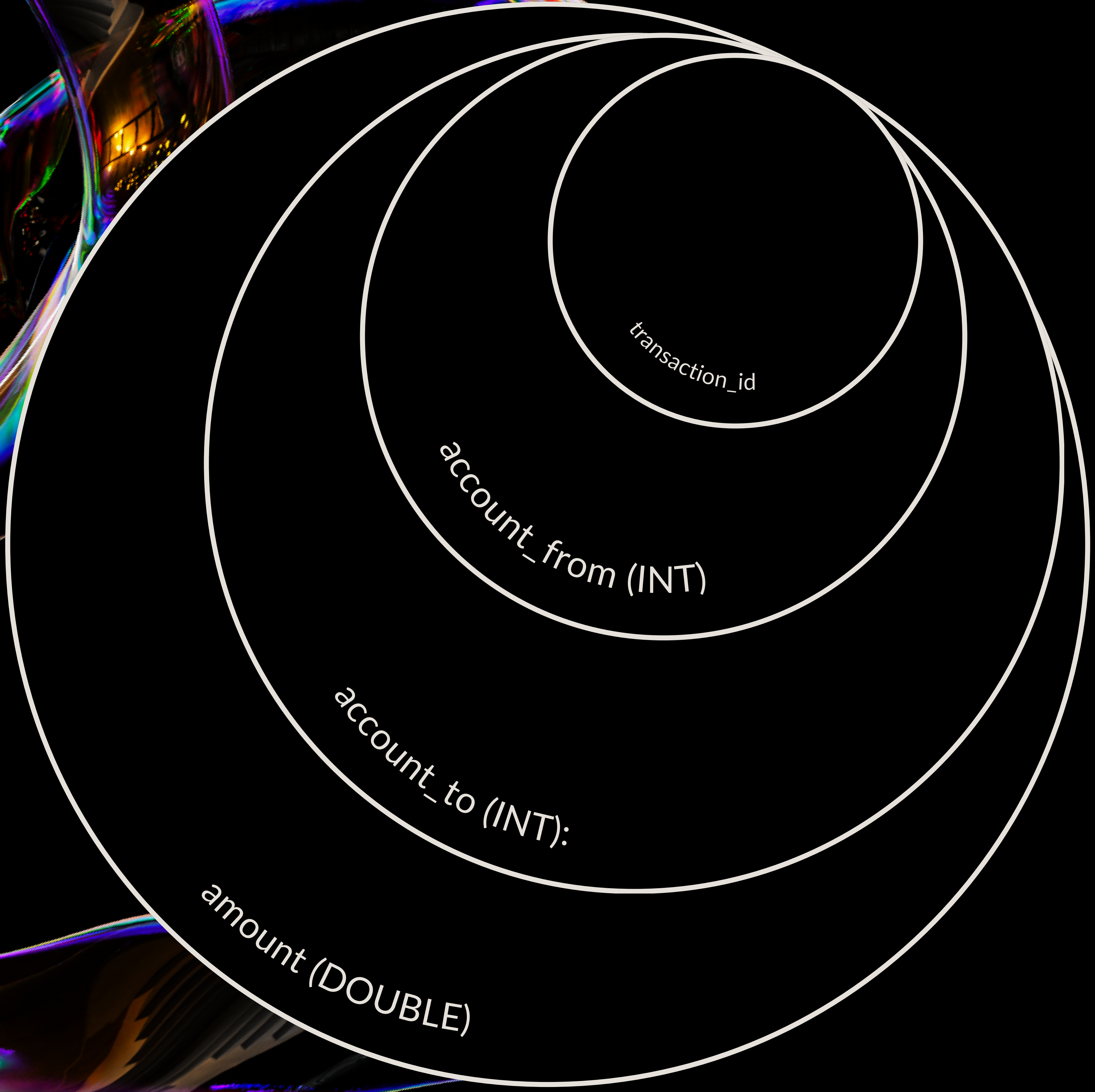
- JDBC (Java Database Connectivity)
- Swing

DATABASE SCHEMA

Accounts Table:

- account_number (INT): Unique identifier for each account.
- name (VARCHAR): Name of the account holder.
- password (VARCHAR): Password for secure login.
- balance (DOUBLE): Account balance.





transaction_id

account_from (INT)

account_to (INT):

amount (DOUBLE)

DATABASE SCHEMA

Transactions Table:

- transaction_id (INT, AUTO_INCREMENT): Unique transaction identifier.

- account_from (INT): Sender's account number.

- account_to (INT): Recipient's account number.

- amount (DOUBLE): Amount transferred.

- transaction_date (DATETIME): Date and time of the transaction.

FLOW OF THE APPLICATION

ACCOUNT CREATION

Users enter details like name and initial balance, and the system generates a 3-digit account number.

USER LOGIN

Users input their account number and password to log in securely.

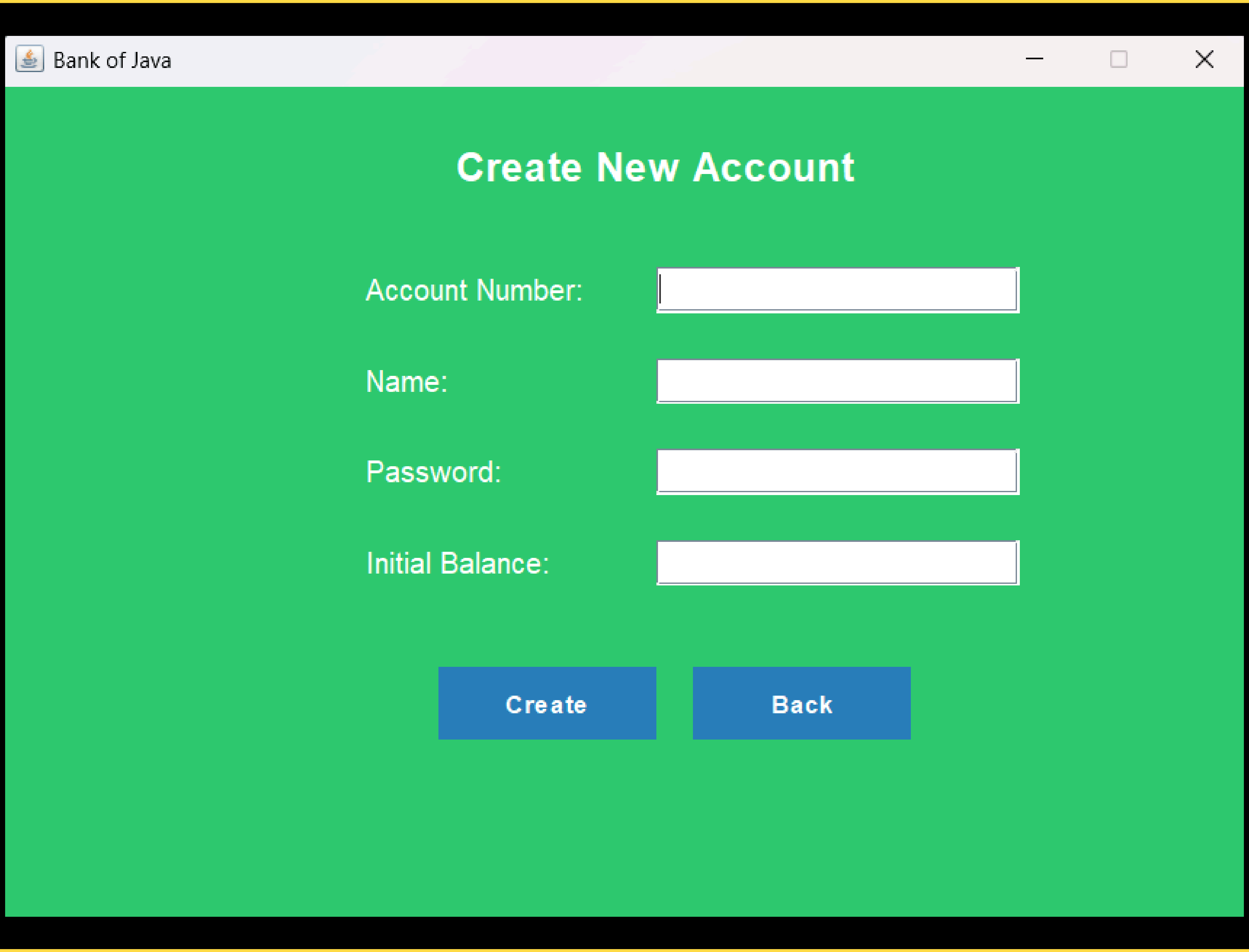
USER PANEL

The User Panel is the core interface for account holders to interact with their banking accounts.

TRANSACTION HISTORY

Users can view a history of all transactions made or received.

ACCOUNT CREATION



Bank of Java

Create New Account

Account Number:	
Name:	
Password:	
Initial Balance:	

FLOW OF THE APPLICATION

ACCOUNT CREATION

Users enter details like name and initial balance, and the system generates a 3-digit account number.

USER LOGIN

Users input their account number and password to log in securely.

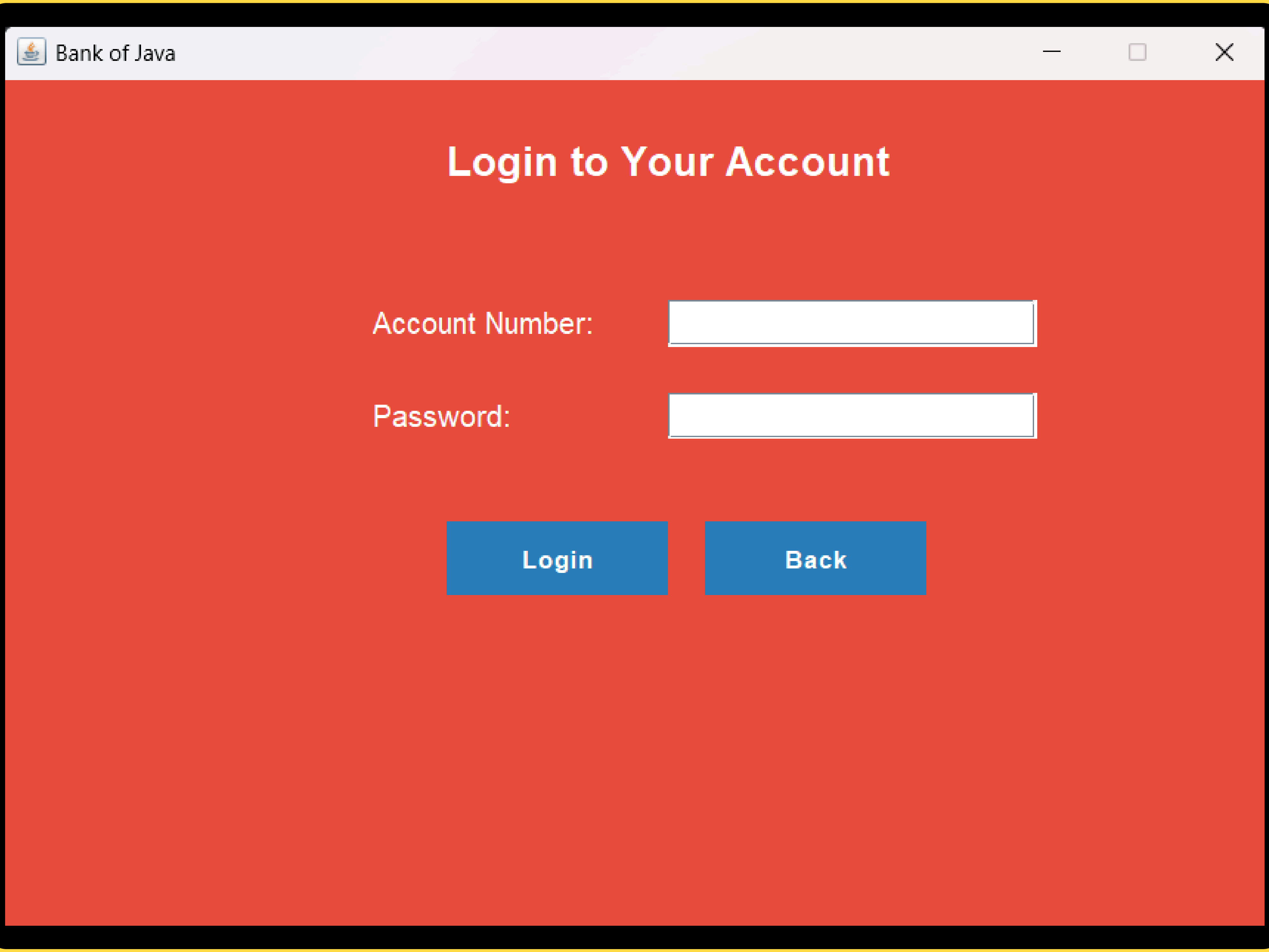
USER PANEL

The User Panel is the core interface for account holders to interact with their banking accounts.

TRANSACTION HISTORY

Users can view a history of all transactions made or received.

USER LOGIN



Bank of Java

Login to Your Account

Account Number:

Password:

FLOW OF THE APPLICATION

ACCOUNT CREATION

Users enter details like name and initial balance, and the system generates a 3-digit account number.

USER LOGIN

Users input their account number and password to log in securely.

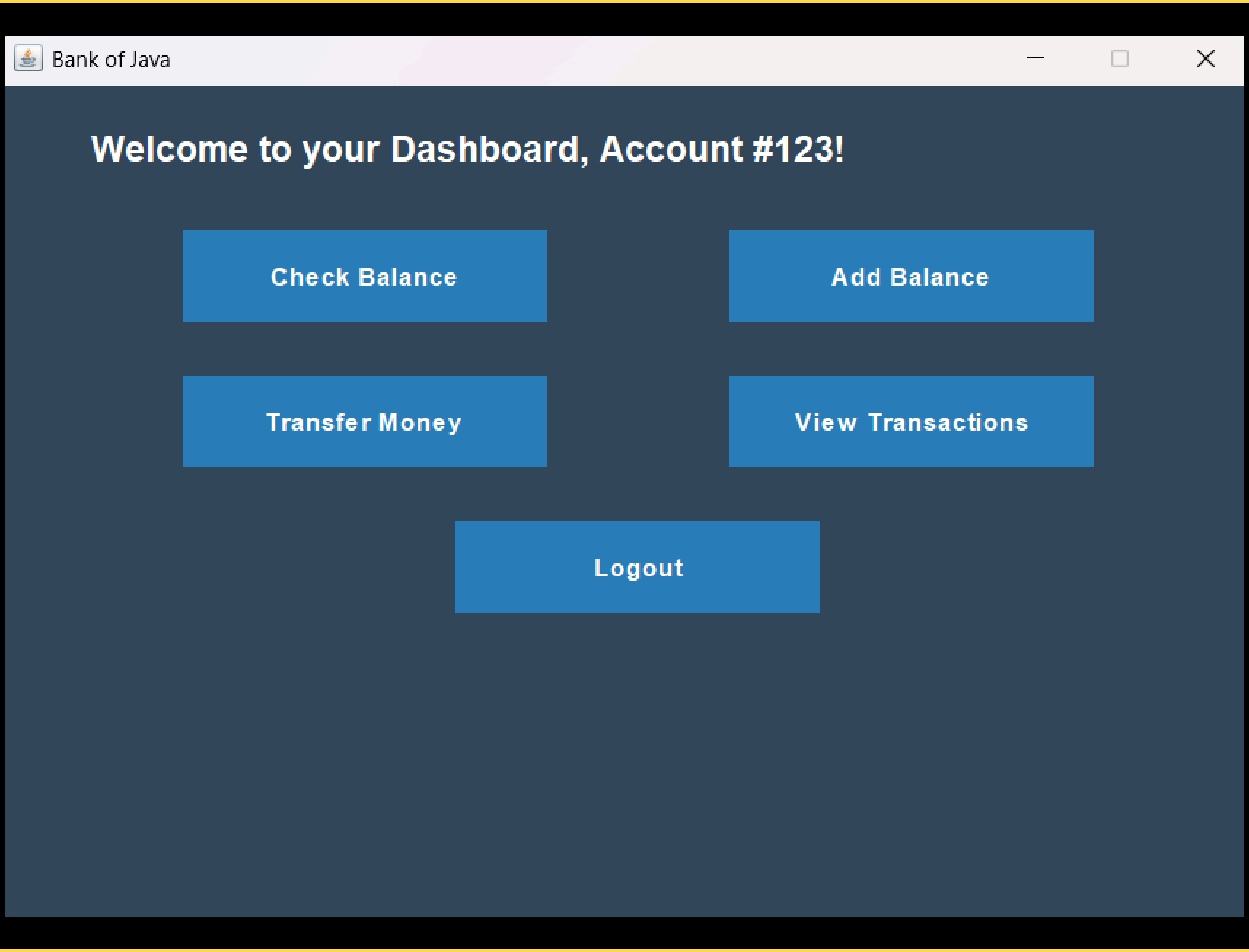
USER PANEL

The User Panel is the core interface for account holders to interact with their banking accounts.

TRANSACTION HISTORY

Users can view a history of all transactions made or received.

USER PANEL



Welcome to your Dashboard, Account #123!

FLOW OF THE APPLICATION

ACCOUNT CREATION

Users enter details like name and initial balance, and the system generates a 3-digit account number.

USER LOGIN

Users input their account number and password to log in securely.

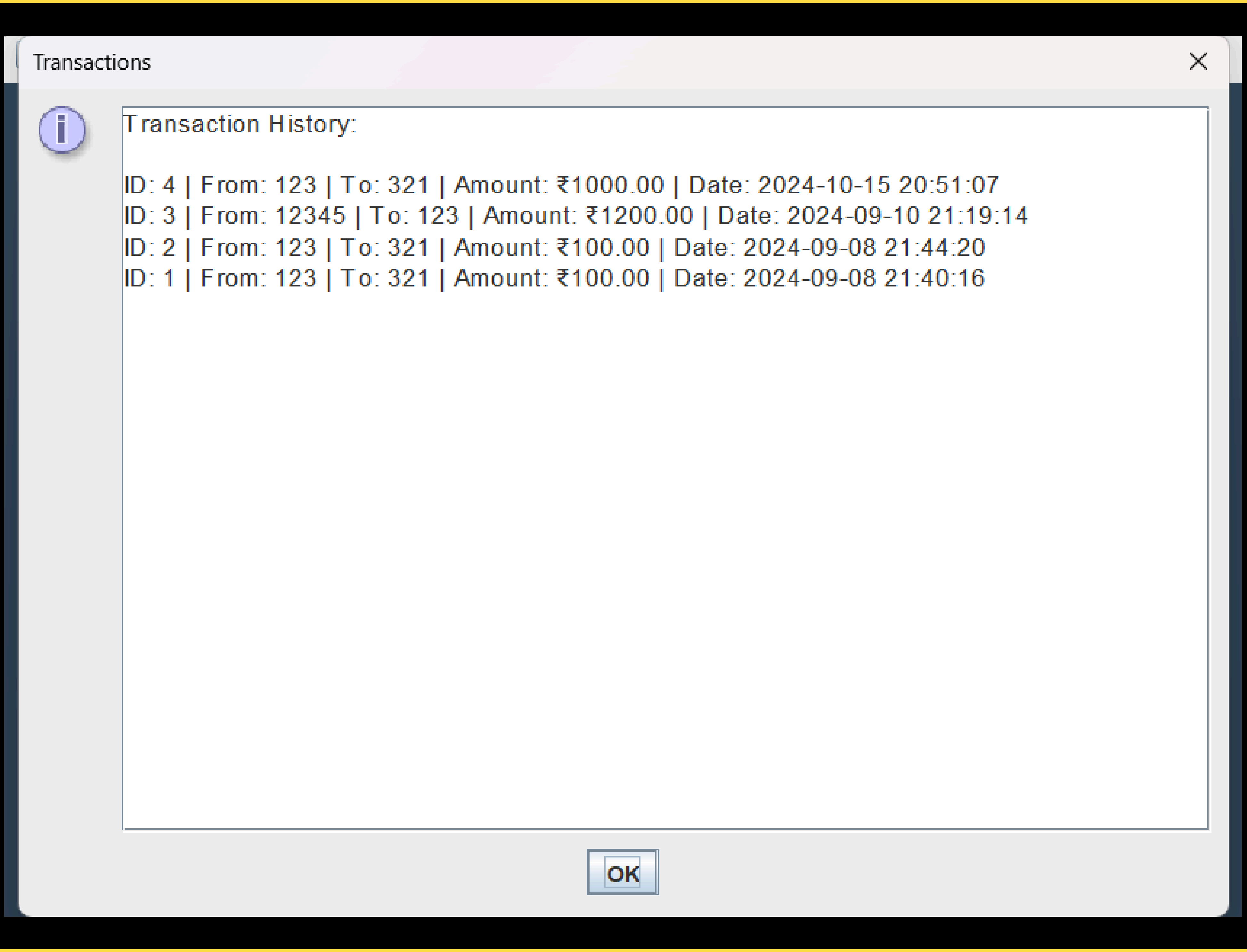
USER PANEL

The User Panel is the core interface for account holders to interact with their banking accounts.

TRANSACTION HISTORY

Users can view a history of all transactions made or received.

TRANSACTION HISTORY



```
Transaction History:

ID: 4 | From: 123 | To: 321 | Amount: ₹1000.00 | Date: 2024-10-15 20:51:07
ID: 3 | From: 12345 | To: 123 | Amount: ₹1200.00 | Date: 2024-09-10 21:19:14
ID: 2 | From: 123 | To: 321 | Amount: ₹100.00 | Date: 2024-09-08 21:44:20
ID: 1 | From: 123 | To: 321 | Amount: ₹100.00 | Date: 2024-09-08 21:40:16
```

CHALLENGES FACED

JDBC Integration

Establishing a seamless connection between the Java application and MySQL database.

Data Security

Ensuring account information is stored securely and preventing unauthorized access.

Error Handling

Managing errors during transactions and ensuring data integrity.

User Experience

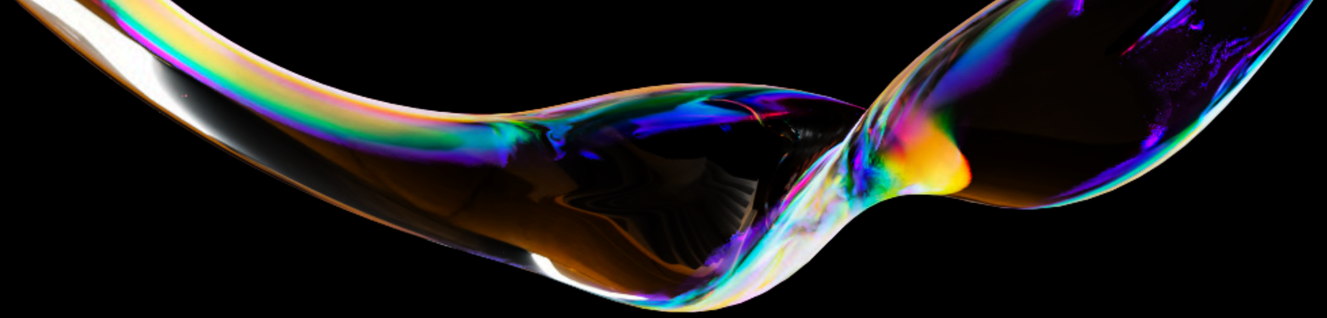
Designing a simple, user-friendly interface for a smooth user experience.

CODE - GITHUB REPOSITORY

Scan the QR Code to Visit the GitHub Repository

- This repository contains the full source code for our Banking Application using JDBC project.
- The code is well-documented and organized for easy navigation.
- Contributions and improvements are welcome!

https://github.com/YashGarg1122/BankOfJava.git



CONCLUSION

- This project successfully achieved the objective of creating a secure and functional Banking Application using JDBC.
- The system allows users to create accounts, manage balances, and perform transactions efficiently.
- Our team gained valuable experience in JDBC, database management, and error handling, contributing to real-world software development skills.

TEAM CONTRIBUTIONS

Thank you for your time! Reach out to us for questions.

YASH GARG

Lead developer, account management.

DHRUV GUPTA

Transaction handling, JDBC integration

APURVA BANSAL

Error handling, testing, database design

RITESH KUMAR SINHA

Interface design, project management

ANSHUL SIROHI

Documentation, flow diagrams

SAURAV AGRAWAL

Git management, database setup

THANK YOU

for your time and attention