

An abstract graphic on the left side of the slide, featuring a dark blue background with a lighter blue diagonal band. Along this band are several white icons: a padlock, a document with a checkmark, a server rack, a document with a magnifying glass, and a document with a list. There are also some geometric shapes like circles and triangles scattered around.

Introduction to SupperBank

SupperBank: Desktop Banking Application

Zakariya SABRI && Youssef AOUZAH

UM6P College of Computing Rabat

Quick Project Overview

For Clients:

- Deposit money
- Withdraw money
- Transfer funds
- View transaction history
- Convert currencies
- Update profile

For Administrators:

- Manage all clients
- View all transactions
- Monitor bank statistics
- Manage currency rates
- Add/Edit/Delete clients
- Search clients

TECHNOLOGY STACK

Java

Java Swing

MySQL

JDBC

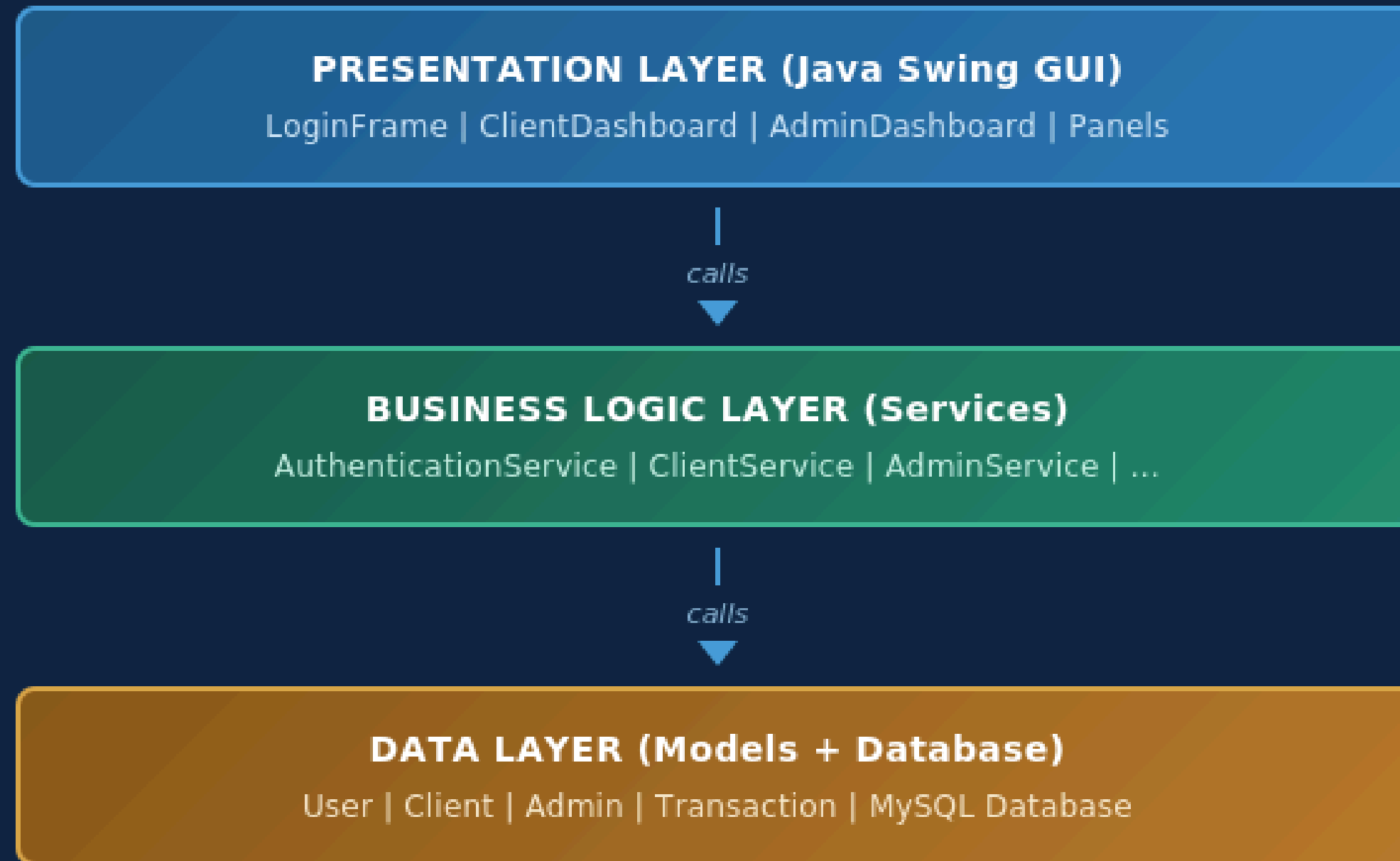
Why SupperBank?

- **OOP friendly project**
- **Fundamental level java project**
- **Helps understanding how banking systems work**

What We Wanted to Learn:

Challenge	What We Learned
How do banks handle multiple users?	Role-based authentication
How are transactions kept safe?	Database transactions (COMMIT/ROLLBACK)
How do GUIs talk to databases?	3-Layer Architecture
How to build professional apps?	Java Swing + Design Patterns

System Architecture - 3-Layer Architecture



Why 3 Layers?

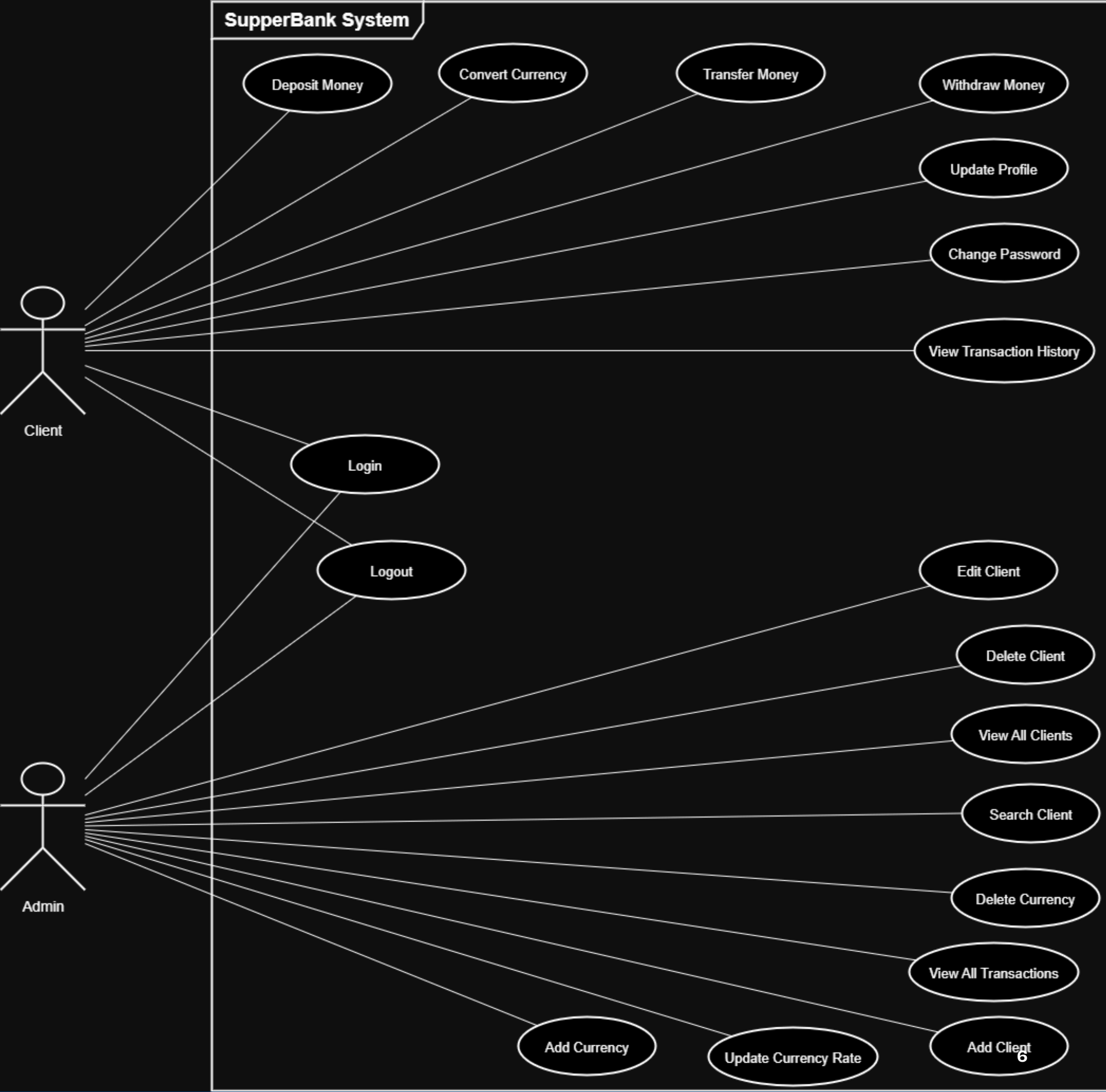
Separation of Concerns

Easy to Modify

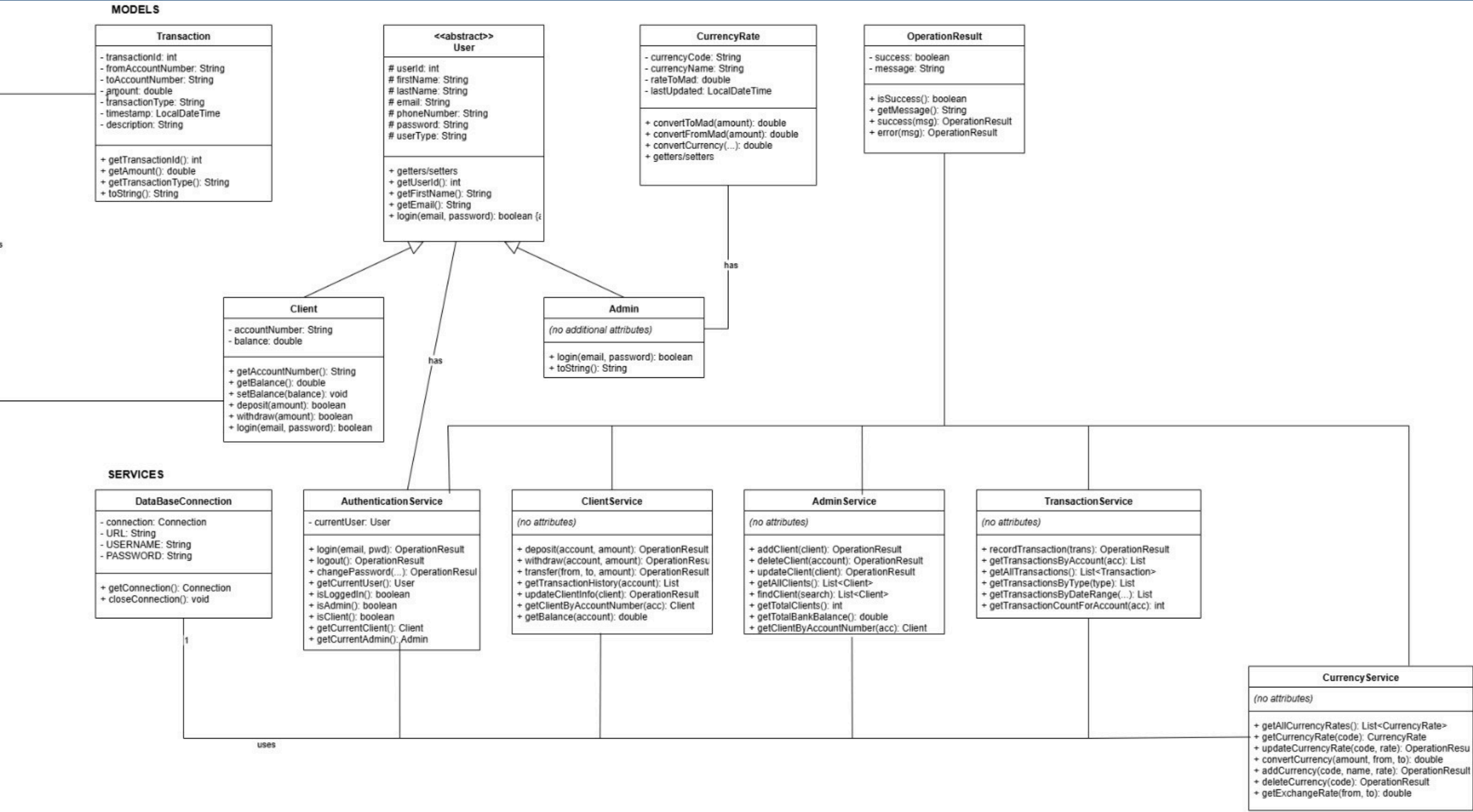
Easy to Test

Team Work

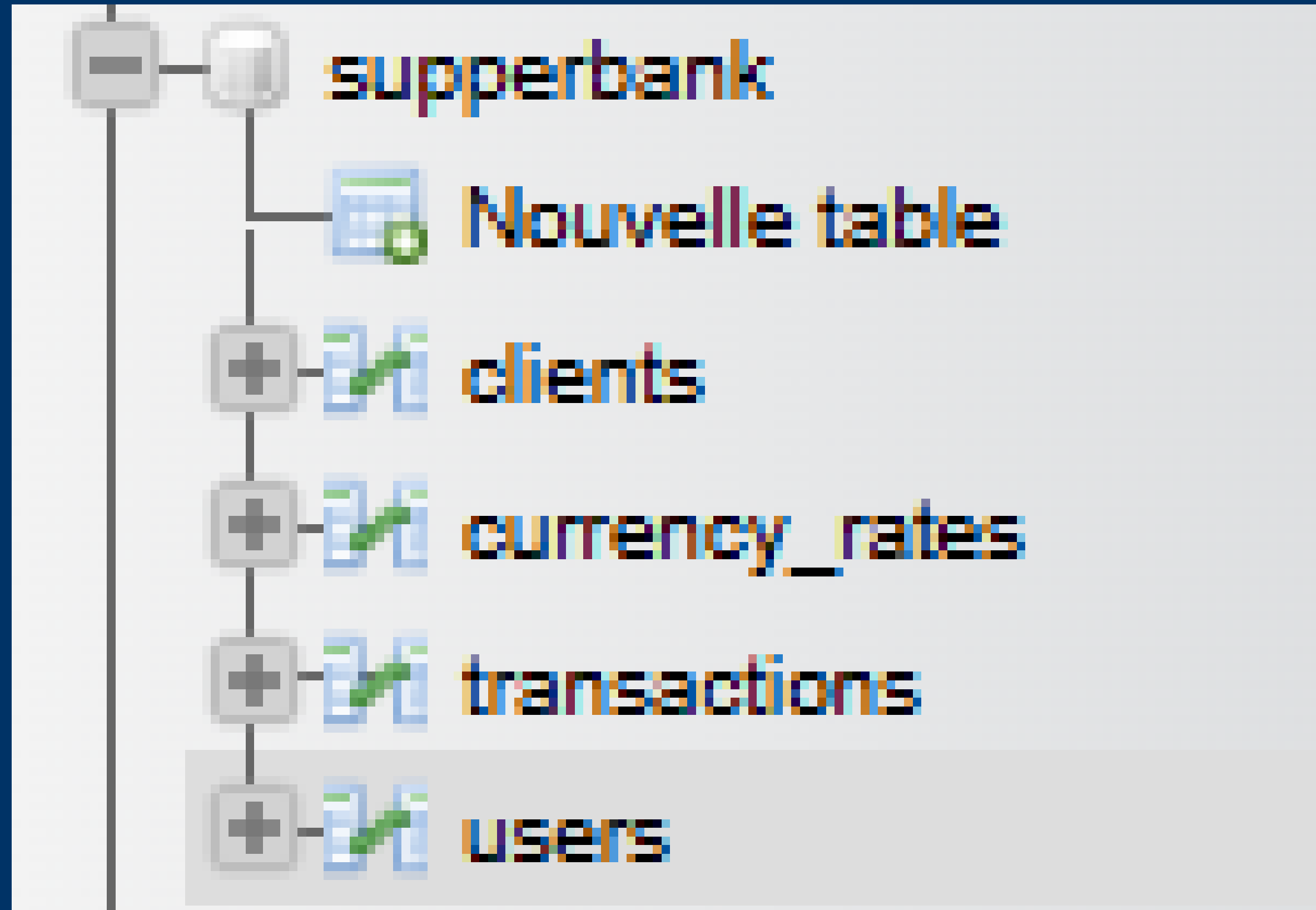
Use Case Diagram



Class Diagram



Database Schema Overview



NOTES

DATABASE STRUCTURE ENSURES
EFFICIENT DATA MANAGEMENT.

Client-Database Architecture

client_id	account_number	balance
3	AA08	80.00

currency_rates table Architecture

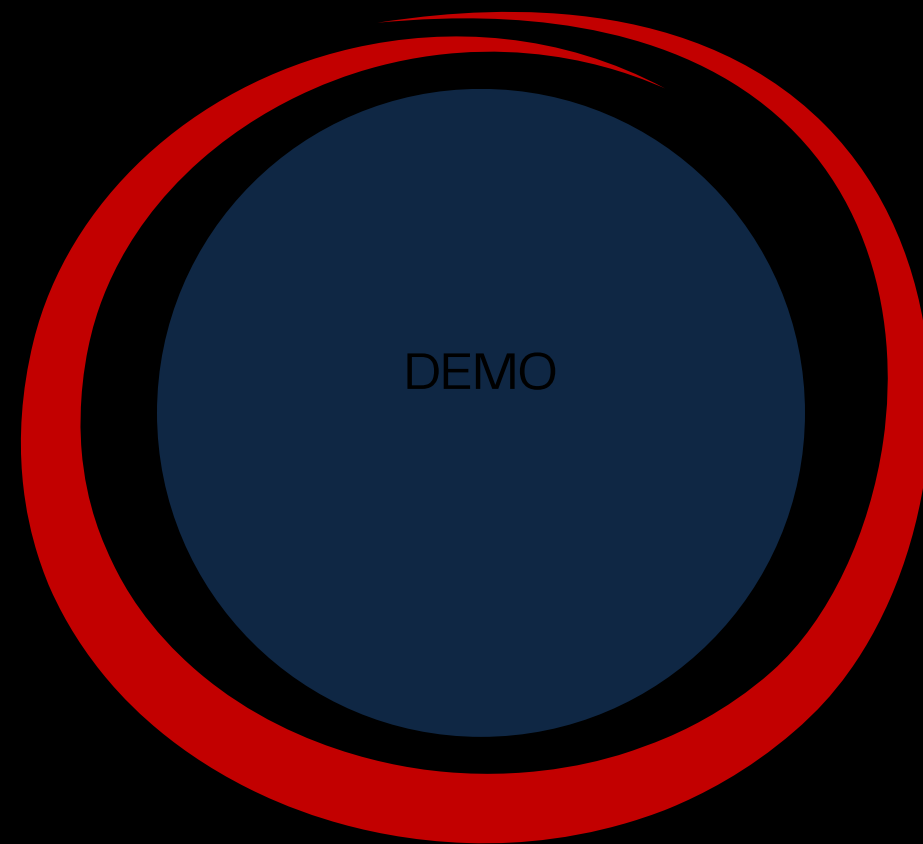
currency_code	currency_name	rate_to_mad	last_updated
CAD	Canadian Dollar	1.350000	2025-11-29 10:23:15

transaction table Architecture

transaction_id	from_account_number	to_account_number	amount	transaction_type	timestamp	description
1	A77	AA08	11.00	TRANSFER	2025-10-16 01:58:42	Transfer to account AA08

user table Architecture

user_id	first_name	last_name	email	phone_number	password	user_type	created_at
1	Bank	Admin	admin@supperbank.com	+1234567890	admin123	ADMIN	2025-10-15 10:23:14



Design patterns implemented

Singleton design

```
public class DataBaseConnection { 28 usages  👤 zakariya *  
    public static Connection getConnection() { 28 usages  👤 zakariya *  
  
        try{  
            if (_connection == null || _connection.isClosed()) {  
  
                Class.forName( className: "com.mysql.cj.jdbc.Driver");  
  
                _connection = DriverManager.getConnection(_URL, _USERNAME, _PASSWORD);  
  
            }  
            return _connection;  
  
        }catch(SQLException | ClassNotFoundException e){  
            System.out.println(e.toString());  
        }  
        return null;  
    }  
}
```

Factory design

```
1  > import ...
4
5  ▶ public class Main {  ⚡ zakariya *
6
7  ▶     public static void main(String[] args) {  ⚡ zakariya *
8
9
10     try {
11         UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
12     } catch (Exception e) {
13
14         System.out.println("Could not set system look and feel");
15     }
16
17
18
19     SwingUtilities.invokeLater(new Runnable() {  ⚡ zakariya *
20         @Override  ⚡ zakariya *
21         ▶ public void run() {
22
23             LoginFrame loginFrame = new LoginFrame();
24
25
26             loginFrame.setVisible(true);
27         }
28     });
29
30
31 }
```

Observer design

```
// Add hover effect
button.addMouseListener(new MouseAdapter() {  @ zakariya
    @Override  @ zakariya
    public void mouseEntered(MouseEvent e) {
        if (button != currentButton) {
            button.setBackground(SIDEBAR_HOVER);
        }
    }
}
```

Proposed Future Improvements

A database backup system

Mobile App

Enhanced Security

Summary

- GUI implemented
- Design pattern/architecture applied
- SQL database
- Layered architecture
- SOLID principles

Thank You !



Github repository