Java GUI Banking Application Documentation

# Overview

This is a standalone Java Banking Application built using Swing and AWT for the GUI and integrated with a PostgreSQL database via Docker. It follows the Model-View-Controller (MVC) design pattern for clean separation of concerns, modular development, and scalability.

# Features

- User Registration & Login  
- Deposit, Withdrawal, and Transfer Funds  
- View Transaction History  
- Admin Dashboard for Account Management  
- Error Handling with Dialog Feedback  
- Persistent Database Integration (PostgreSQL via Docker)  
- Modular MVC Code Structure

# User Manual

1. Launching the App:  
 - Ensure Docker is running and PostgreSQL container is active.  
 - Run the Java application via the main class BankApp.java.  
  
2. Registering an Account:  
 - Navigate to Sign Up screen.  
 - Enter required details (name, email, password, account type).  
 - Submit to create an account.  
  
3. Logging In:  
 - Navigate to Login screen.  
 - Enter valid credentials to access the user dashboard.  
  
4. User Dashboard:  
 - Balance View, Deposit, Withdraw, Transfer Funds, Transaction History, Logout.  
  
5. Admin Panel:  
 - View users, reset/delete accounts, monitor transactions.

# Technical Documentation

Class Structure:  
- controller: AuthController, TransactionController, AdminController  
- model: User, Account, Transaction, DatabaseManager  
- view: LoginView, SignupView, DashboardView, AdminView  
- util: Validator, DialogUtil, Constants  
- Main: BankApp.java  
  
Database Tables:  
- User Table: id, full\_name, email, password, account\_number, balance  
- Transaction Table: id, user\_id, type, amount, timestamp  
  
GUI Hierarchy (DashboardView):  
- JFrame (DashboardView)  
 - Header (JLabel)  
 - Balance Panel (JLabel)  
 - Action Panel (Buttons: Deposit, Withdraw, Transfer)  
 - JTable (Transaction History)  
  
Exception Handling:  
- Input validations with dialog feedback.  
- Database errors handled via try-catch.  
- Centralized DialogUtil for critical issues.  
  
Threading Model:  
- Single-threaded with GUI on Event Dispatch Thread (EDT).  
- SwingUtilities.invokeLater used for thread safety.  
  
Docker Setup:  
docker run --name banking-db -e POSTGRES\_PASSWORD=pass123 -e POSTGRES\_USER=bankuser -e POSTGRES\_DB=bankapp -p 5432:5432 -d postgres