

BANKING APPLICATION

ABSTRACT:

The Banking Application is a secure and user-friendly software system designed to streamline and digitize core banking operations for customers. It allows users to perform essential banking tasks such as account creation, balance inquiry, money deposit, withdrawal, and transaction history tracking. The application is built with a focus on usability, data integrity, and security, ensuring that all financial transactions are encrypted and access-controlled. For administrators, the application provides backend tools for managing user accounts, monitoring transactions, and generating reports. The system can be developed using technologies such as Python(Tkinter), with integration to relational databases like MySQL for secure data storage. This project aims to replicate the functionality of real-world banking systems in a simplified environment suitable for academic or prototype purposes. It demonstrates how financial institutions can use digital platforms to enhance user experience and operational efficiency.

FUNCTIONAL REQUIREMENTS

Graphical User interface with the User.

Operating Systems supported

1. Windows
2. Linux

Technologies and Languages used to Develop : .Python

Debugger and Emulator : Python Debugger

SOFTWARE REQUIREMENTS

Operating system : Windows 10.

Coding Language : Python.

User interface : Python.

Data Base : MySQL.

Tool : Visual Studio Code/ Python IDLE.

HARDWARE REQUIREMENTS

For developing the application, the following are required :

Hardware System : Desktop or Laptop

Monitor : Any Standard Monitor

Input Devices : Keyboard, Mouse

RAM : 4 GB (Min)

E -R DIAGRAM:

How the Bank Stores Information

User: Stores personal details of the customer including UserID, Name, Email, Phone, and Password.

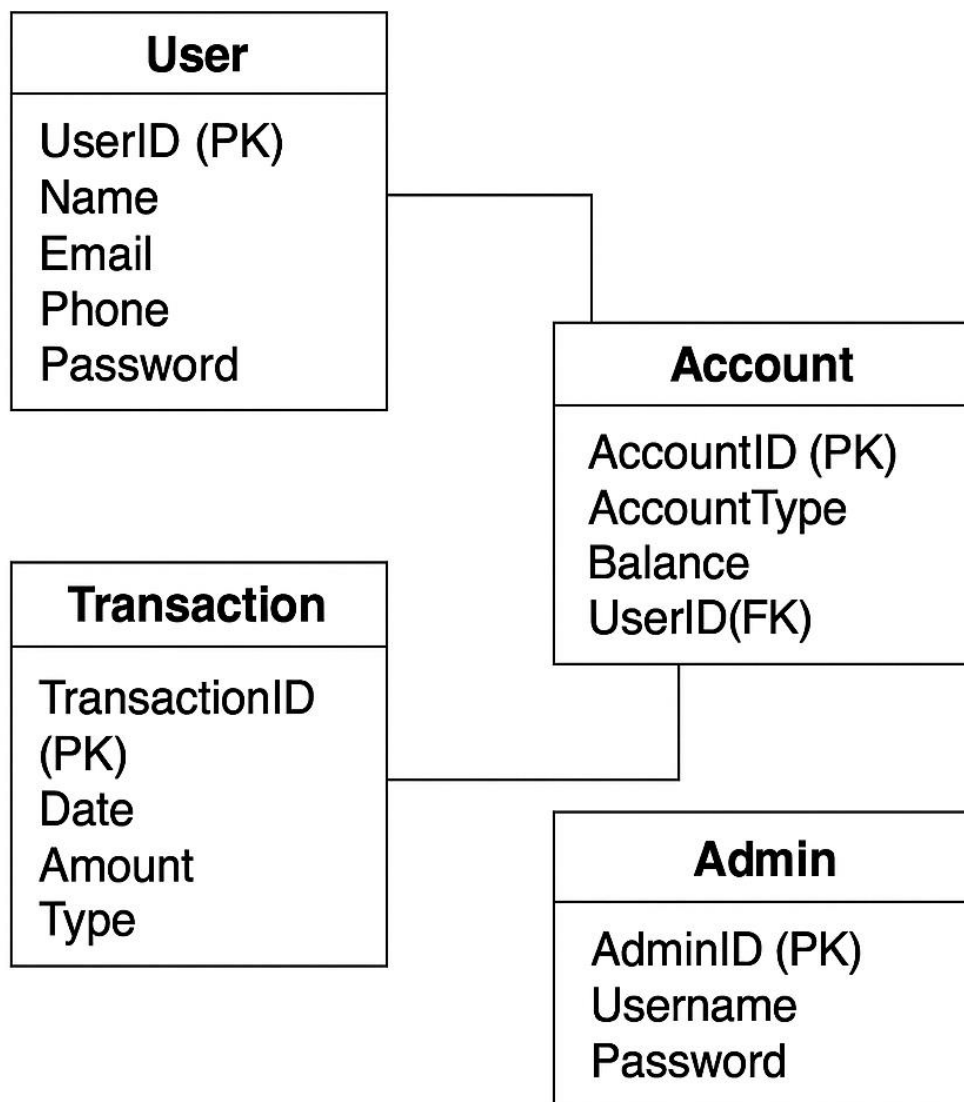
Account: Represents a user's bank account with details like AccountID, AccountType, Balance, and a foreign key UserID linking to the User.

Transaction: Records the details of financial transactions such as TransactionID, Date, Amount, and Type (e.g., deposit or withdrawal).

Admin: Stores admin login credentials with AdminID, Username, and Password for managing the banking system.

User–Account Relationship: One user can have multiple accounts, establishing a one-to-many relationship.

Account–Transaction Relationship: One account can be associated with multiple transactions, forming another one-to-many relationship.



USE CASE Diagram (Admin):

Admin – Represents a system administrator who manages banking operations through the system.

Login – Admin must authenticate with valid credentials to access the system features.

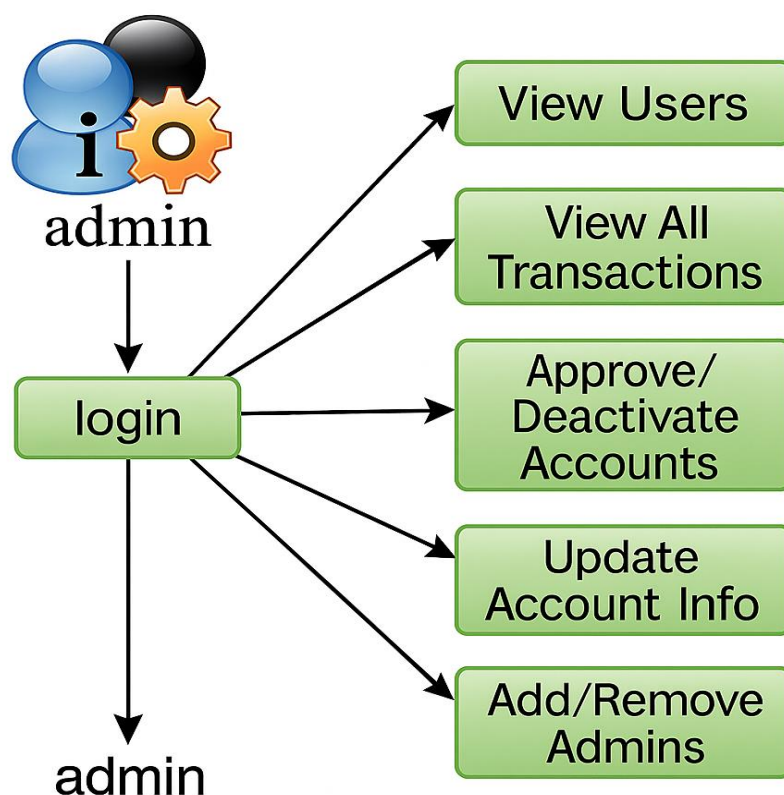
View Users – Admin can view the list of all registered users in the banking system.

View All Transactions – Admin can monitor all user transaction history for oversight and security.

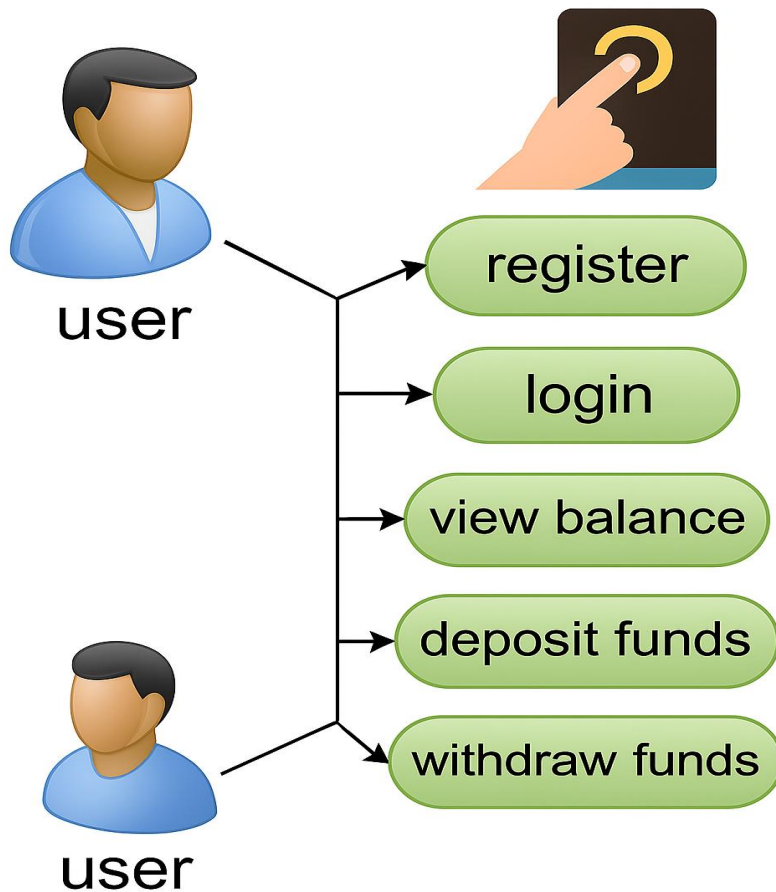
Approve/Deactivate Accounts – Admin has the authority to activate or deactivate user bank accounts.

Update Account Info – Admin can edit and update user account information as necessary.

Add/Remove Admins – Admin can assign or revoke admin privileges to/from other users.



USE CASE Diagram (User):



Register Account – Create a new account with the bank.

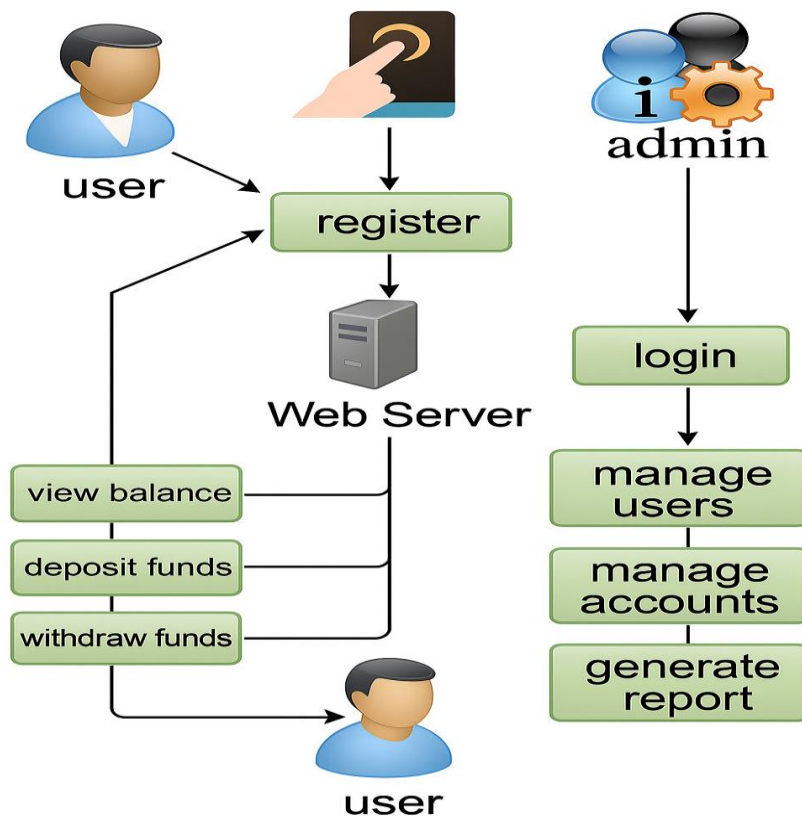
Login (with PIN) – Enter account number and PIN to get access.

View Balance – See how much money they have.

Deposit Money – Put money into their account.

Withdraw Money – Take money out of their account (after PIN check and balance check).

Architecture:



User: A person who can register, log in, and perform banking operations like viewing balance, depositing, and withdrawing funds.

Register: Allows a new user to create an account in the system.

Login (User): Authenticates the user to access their banking features.

View Balance: Enables users to check their current account balance.

Deposit Funds: Lets users add money to their account.

Withdraw Funds: Allows users to remove money from their account.

Admin: A privileged user who can manage the system and oversee all accounts and users.

Login (Admin): Authenticates the admin to access management features.

Manage Users: Lets the admin control user access and information.

Manage Accounts: Enables the admin to handle account-level operations and settings.

Generate Report: Allows the admin to produce reports on system or financial activity.

System Server: Central backend that processes all user and admin requests and stores data securely.