**CONSOLE BASED BANKING APPLICATION**

* By Ravi Agrahari

PROJECT OBJECTIVE:

A console-based banking application to learn and simulate the banking features using concept of JAVA OPPs and fundamentals.

IDENTIFICATION OF NEED:

* + Digitalize the working of Banking.
  + Make banking transactions online and secure.
  + Provide easy and fast access to banking resource to customers.

PLATFORM SPECIFIATION:

HAREWARE REQUIREMENTS:

* + - At least 4 GB Ram
    - 256 SSD
    - Intel core i3 processor

SOFTWARE REQUIREMENTS:

* + - JAVA
    - SQL
    - JDBC
    - NetBeans / Eclipse / IntelliJ IDEA
    - JIRA

FUCTIONAL REQUIREMENTS:

1. User Account Management:

* Saving and current accounts

1. Transaction Management:

* Deposit and withdraw funds
* Transfer funds
* Thread-safe transactions

1. Transaction History:

* Save a logs of all transactions
* Retrieve and display transactions

1. Database Operations (Using JDBC):

* CRUD operation
* Relational database

1. Reports:

* Account details
* Balance across all accounts
* Number of accounts
* Summaries of deposits, withdrawals and transfers

USE CASE DIAGRAM:

A screenshot of a diagram

Description automatically generated

SCHEMA DESIGN:

**Bank**: Stores information about the bank

Columns:

* bank\_id INT,
* bank\_name VARCHAR(100) ,
* bank\_branch VARCHAR(100),

**Account**: Store account details , associated with a particular bank

Columns:

* + account\_id INT UNIQUE(PK),
  + customer\_id INT,
  + bank\_id INT FOREIGN KEY,
  + account\_type VARCHAR(50) ,

**SavingsAccount and CurrentAccount** : Specialized table for different account types , inheriting form the Account table

Columns (Saving Account):

* + account\_id INT (FK)
  + interest\_rate DECIMAL (5,2)

Columns (Current Account):

* + account\_id INT (FK)
  + Overdraft\_limit DECIMAL (15,2)

**Transaction:** Records all transactions (deposit, withdraw, transfer) Linked to accounts.

Columns:

* account\_id INT (FK)
* transaction\_id INT (PK)
* transaction\_type VARCHAR(50)
* amount DECIMAL(15,2)
* transaction\_date TIMESTAMP

**DepositTransaction , WithdrawTransaction:**  Specialized tables for different transactions types, inheriting from the transaction table

Columns ( WithdrawTransaction):

* Transaction\_id INT (FK)
* Withdrawal\_method VARCHAR (50)

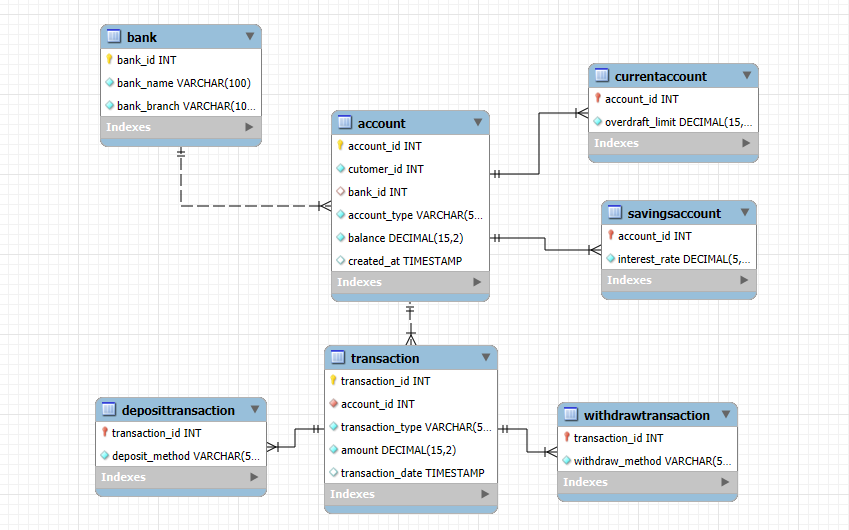
Columns ( DepositTransaction):

* Transaction\_id INT (FK)
* Deposit\_method VARCHAR (50)

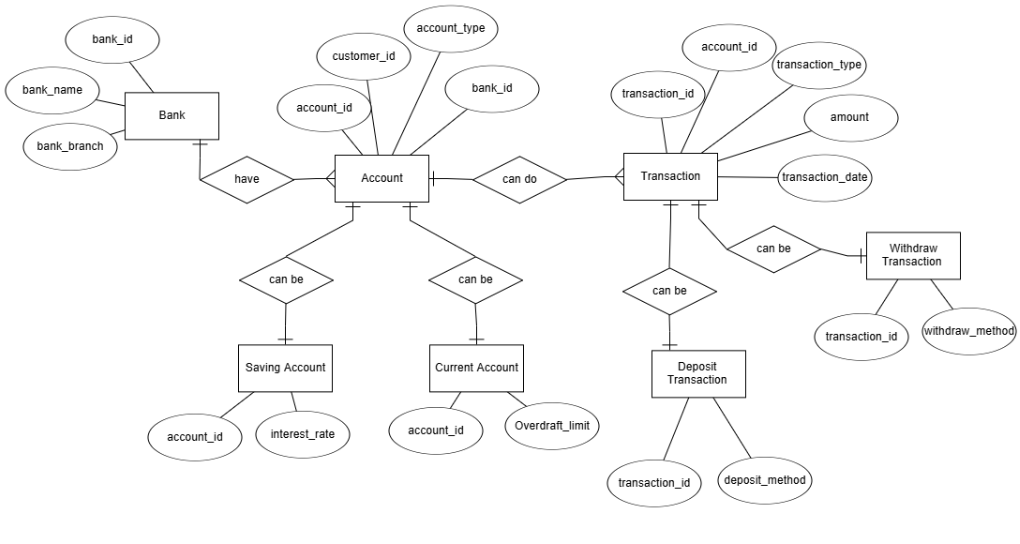
**SCHEMA DIAGRAM:**

A screenshot of a computer

Description automatically generated



**ENITITY-RELATIONSHIP DIGRAM:**

****