**CONSOLE BASED BANKING APPLICATION:**

**Objective:**

To create a console based application has functionalities of creating accounts, managing transactions, viewing account details, and handling database connectivity using Java,DBMS and also Database(SQl) also version control includes using Git

**Identification of Need:**

To build a banking application that provides essential banking functionalities simplicity, security, and data management, enabling users to perform basic banking operations

**Platform Specification:**

**Hardware:**

INTEL CORE I3,I5,I7 OR MAC,MINIMUM 4GB RAM

**SOFTWARE:**

JAVA ,SQL ,JDBC,JIRA

Functional Requirements:

User Account Management:

* Create different types of accounts: Savings and Current. To View account details, such as account type, balance, and other information to Update account information, such as contact details

Transaction Management:

* Deposit and Withdraw funds from accounts.
* Transfer funds between accounts.
* Ensure thread-safe transactions using multithreading and synchronization, so deposits and withdrawals don't conflict.

Transaction History:

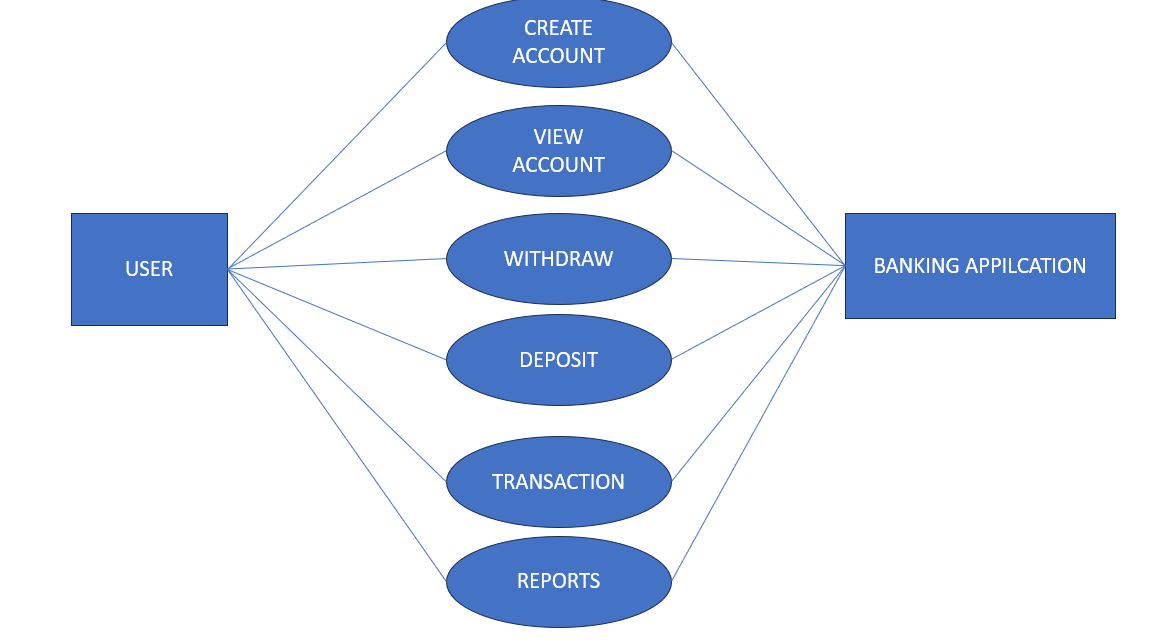
* Save a log of all transactions to a file, allowing users to view a his their account activities.
* Retrieve and display transaction history for specific accounts from saved file.

Database Operation (Using JDBC):

* Store account and transaction details in a relation Perform CRUD operations.

Reports:

USE CASE DIAGRAM:



SCHEMA DESIGN:

**BANK:** Stores information about the bank

Columns:

bank\_id INT UNIQUE (PK)

bank\_name VARCHAR (100)

bank\_branch VARCHAR (100)

**ACCOUNT:** Stores account details, associated with a particular bank

Columns:

Account\_id INT UNIQUE (PK)

Customer\_id INT

Bank\_id INT(FK)

Account\_type VARCHAR (100)

**SAVINGS ACCOUNT AND CURRENT ACCOUNT:** Specialized tables for different Types, inheriting from the Account Table

Columns (SavingsAccount):

Account\_id INT(FK)

Interest\_rate DECIMAL (5,2)

Columns (CurrentAccount):

Account\_id INT(FK)

Overdraft\_limit DECIMAL (15,2)

**TRANSACTION:**Records all transaction(deposit, withdraw, transfer) Linked accounts

Columns (SavingsAccount):

Transaction\_id INT(PK)

Account\_id INT(FK)

Transaction\_type VARCHAR (50)

Amount DECIMAL (15,2)

Transation\_date TIMESTAMP

**DepositTransaction, WithdrawTransaction**: Specialized tables for different Transaction types, inheriting from the Transaction Table

WithdrawTransaction:

Transaction\_id INT(FK)

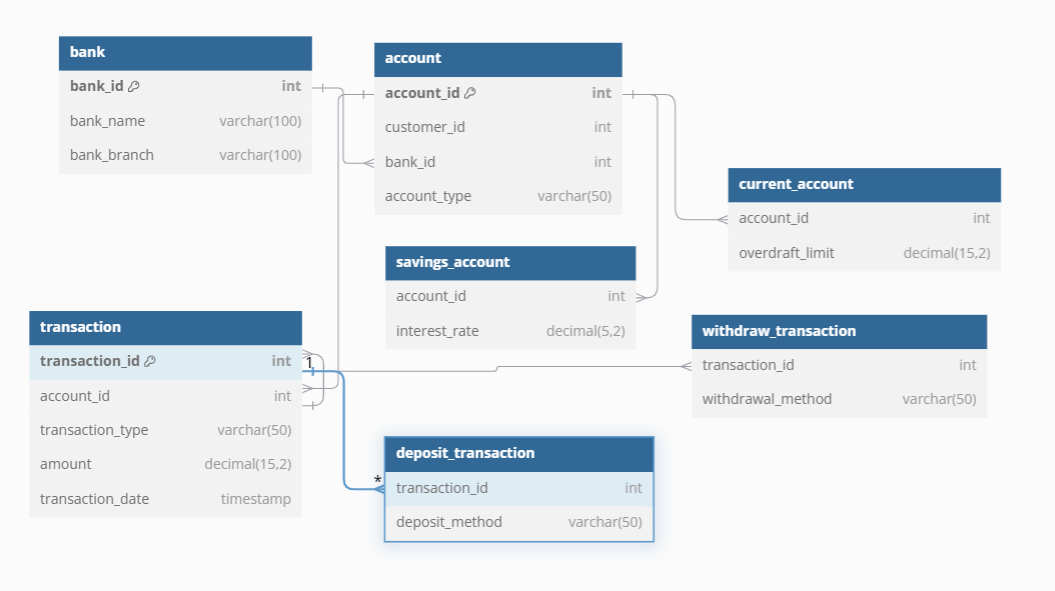
Withdrawal\_method VARCHAR (50)

DepositTransaction:

Transaction\_id INT(FK)

Deposit\_method VARCHAR (50)

SCHEMA DIAGRAM:



ER DIAGRAM:

