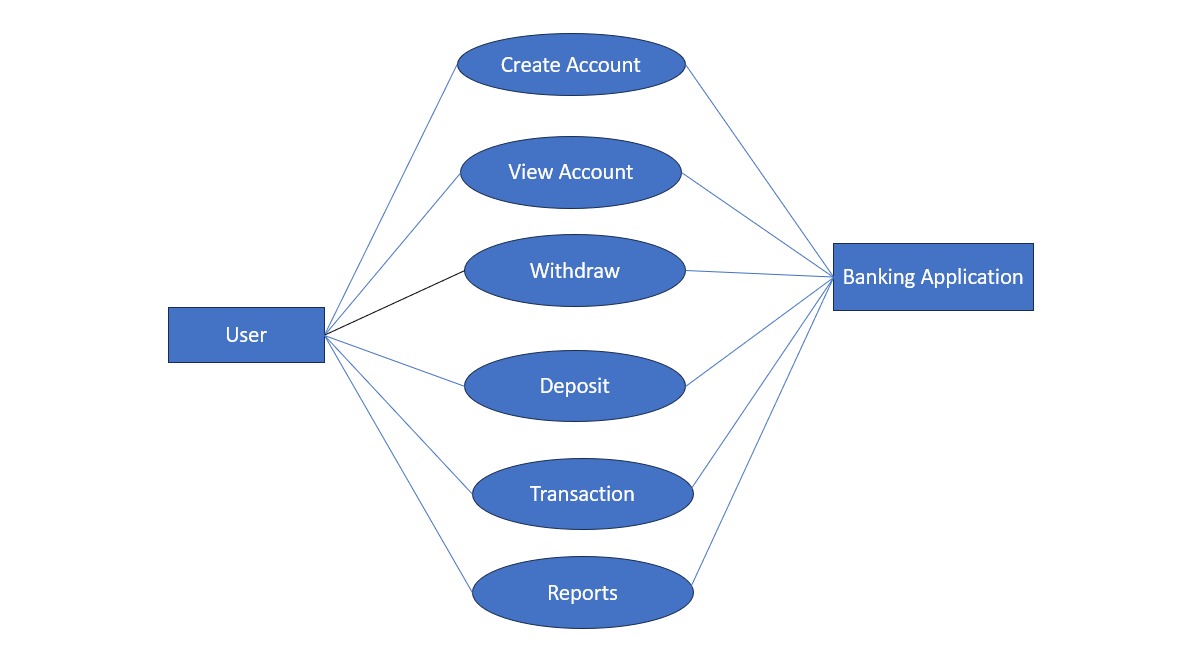
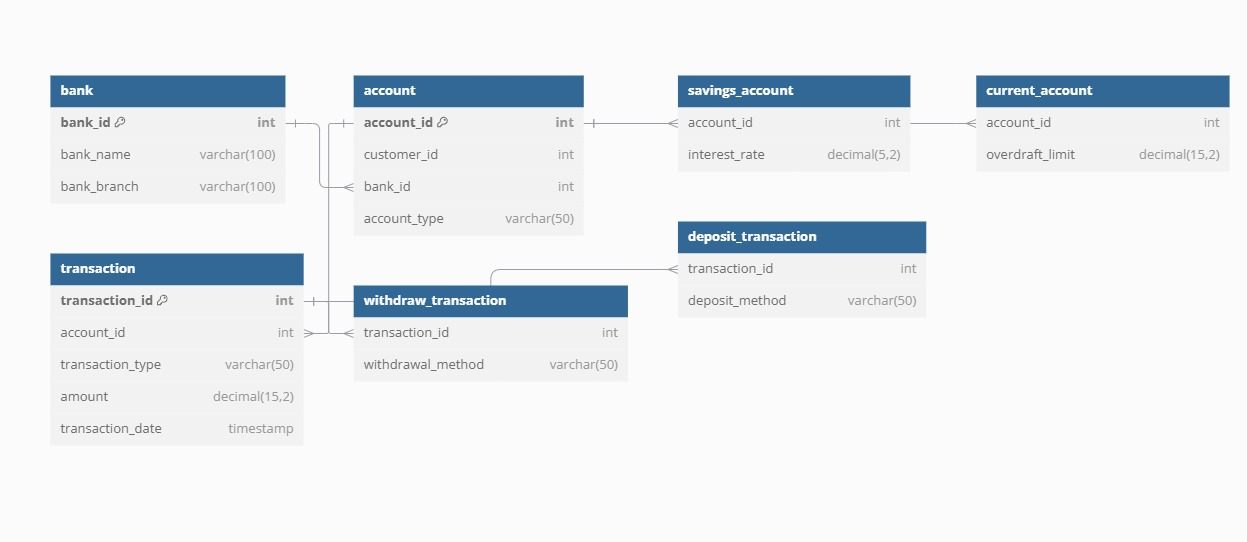
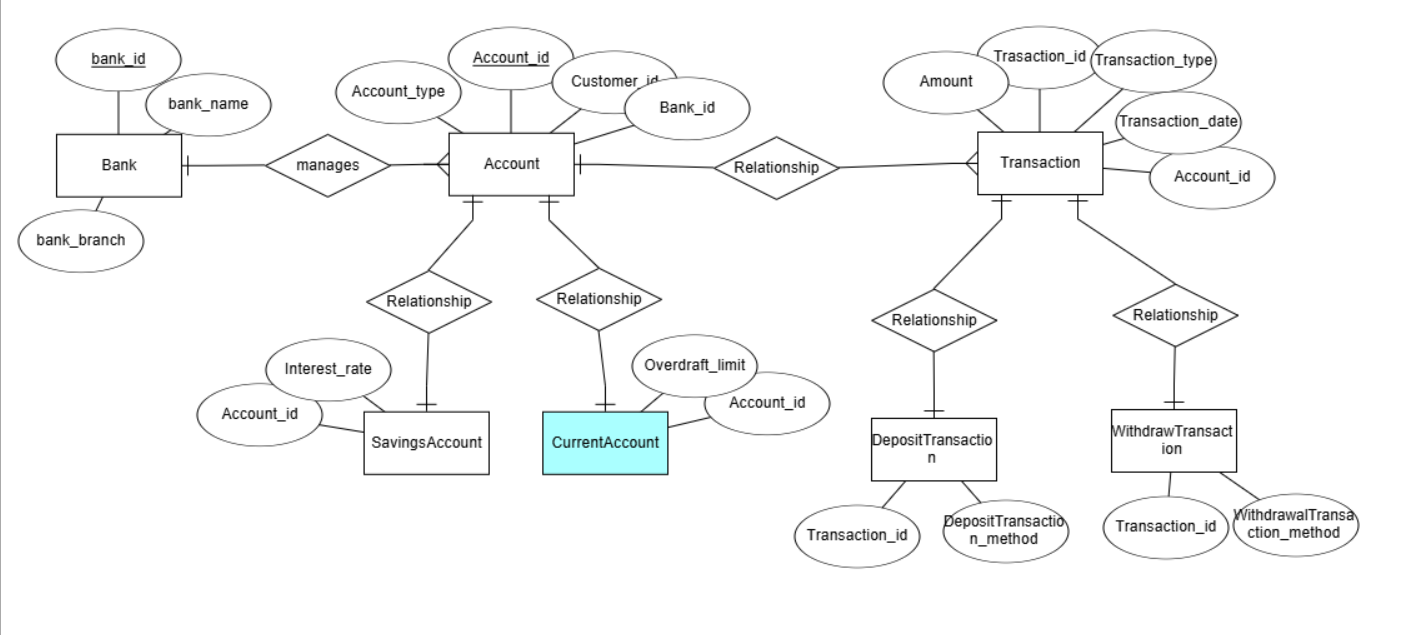
* **CONSOLE BASED BANKING APPLICATION**  
    
  **OBJECTIVE :**   
   Develop a console-based banking application using Java and SQL to manage account and transaction operations efficiently. This application provides secure, thread-safe transaction handling and essential banking functions, enhancing accessibility and ease of use for customers and bank staff.  
    
  **IDENTIFICATION OF PROJECT :**  
   This project serves the banking sector by offering a lightweight tool for account management, transaction tracking, and report generation. It improves operational efficiency, enabling both customers and bank staff to manage accounts and view transaction history seamlessly.  
     
    
  **PLATFORM SPECIFICATION:**  
  **HARDWARE** : INTEL CORE I7,I8,I9 OR MAC OR LINUX,MINIMUM 4GB,SSD OR NVM STORAGE  
     
  **SOFTWARE** : JAVA,SQL,JDBC,JIRA  
    
  **FUNCTIONAL REQUIREMENTS** :  
  1. User Account Management
* o Create different types of accounts: Savings and Current.
* o View account details, such as account type, balance, and other information.
* o Update account information, such as contact details.
* 2. Transaction management:
* o Deposit and Withdraw funds from accounts.
* o Transfer funds between accounts.
* o Ensure thread-safe transactions using multithreading and synchronization, so deposits and withdrawals don’t conflict.
* 3. Transaction History:
* o Save a log of all transactions to a file, allowing users to view a history of their account activities.
* o Retrieve and display transaction history for specific accounts from the saved file.
* 4. Database Operations (Using JDBC):
* o Store account and transaction details in a relational database.
* o Perform CRUD operations for account management, ensuring data persistence.
* o Generate reports based on the data store in the database.
* 5. Reports
* o Generate various reports, such as:
* • Account details for each customer.
* • Total balance across all accounts.
* • Number of accounts by type (savings or current).
* • Daily summaries of deposits, withdrawals and transfers.

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(50)  
  
**Savings : Account and CurrentAccount:** Specialized tables for different account  
 account types, inheriting from thee Account table.  
  
Columns(SavingsAccount)  
Account\_id INT(FK)  
Interest\_rate DECIMAL(5,2)  
  
Columns(CurrentAccount)  
Account\_id INT(FK)  
Overdarft\_limit DECIMAL(15,2)  
  
  
**TRANSACTION** **:** Records all transactions (deposist,withraw,transfer) Linked to accounts.  
  
Cloumns:  
Transaction\_id INT(PK)  
Account\_id INT(FK)  
Transaction\_type varchar(50)  
Amount DECIMAL (15,2)  
Transaction\_date TIMESTAMP  
  
DeposistTransaction, WithdrawTransaction: Specialized tables for Different transaction types, inheriting from the transaction table  
  
WithdrawTransaction:  
Transaction\_id (FK)  
Withdrawal\_method varchar(50)  
  
DepositTransaction:  
  
Transaction\_id (FK)  
Deposit\_method varchar(50)  
  
  
**Schema Diagram :** **Entity-Relation Diagram:  
  
  
**