ATM SIMULATION

DESCRIPTION

A console-based Java application simulating core ATM functionalities, including PIN verification, balance checking, depositing money, withdrawing money, changing PIN, generating a mini statement, and exiting the system. The project mimics real-world ATM operations for a single user account, focusing on secure transactions and user-friendly interaction.

PROBLEM STATEMENT

Manual banking processes are time-consuming and prone to errors. ATMs provide 24/7 access to basic banking operations, building a simulation helps understand banking logic, security, and transaction handling and also initiating from the simple console to a Gui. This project simulates an ATM to practice OOP, exception handling, and transaction management in a controlled environment.

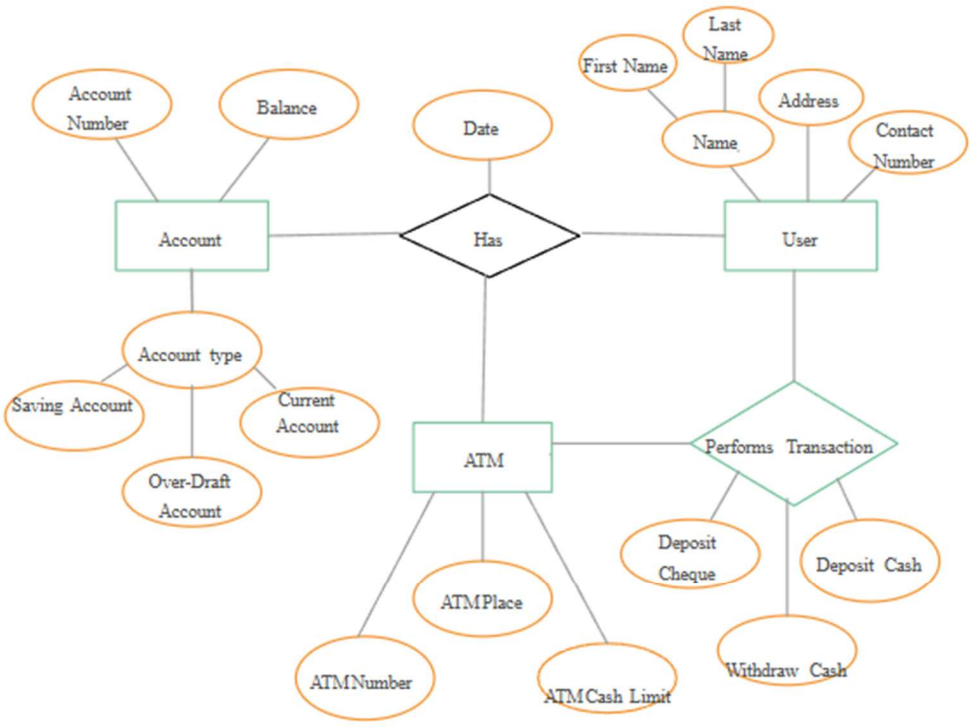
TARGET USERS

Bank Customers: Individuals who use ATMs for quick, self-service banking tasks such as withdrawing cash, depositing money, checking account balances, or reviewing recent transactions. These users expect a secure, user-friendly interface to manage their accounts without visiting a bank branch.

Bank Staff: Banking professionals or software developers prototyping ATM software to test transaction logic, security features (like PIN verification), or user experience before deploying real ATMs.

Educational Users: Students or instructors in programming courses using the project to learn or teach OOP principles, exception handling, and transaction management in a simulated banking environment.

CLASS DIAGRAM



OOP CONCEPTS AND THEIR USES

FUNCTION	OOP CONCEPTS USED	APPLICATION
ATM(constructor)	Constructor,Encapsulation	Initialize private fields
Verify pin()	Encapsulation,Data Hiding	Private pin access
checkBalance()	Encapsulation,DataHiding	Checks Balance
deposit()	Encapsulation,Data Hiding	Update Balance
changePin()	Encapsulation,Data Hiding	Modify Pin
miniStatement()	Encapsulation,Abstraction	Retrieves Mini Statement
showMenu()	Abstraction	Present simplified interface
main()	Classes and Objects	Create ATM object