

EXPERIMENT 4

AIM:

To design a **UML Diagram** for an **ATM System** that models the interactions between the **Customer, Banking System, Administrator, and Technicians** in performing financial transactions and maintaining the system.

PROCEDURE:

1. Identify Main Actors

- **Customer:** Interacts with the ATM to perform financial transactions.
- **Banking System:** Processes the customer's requests and manages account data.
- **Administrator:** Manages ATM settings and monitors transactions.
- **Technician:** Maintains and repairs the ATM hardware and software.

2. Define Use Cases (Functionalities)

Customer Actions:

- **Withdraw Cash:** Customer inputs account details and withdraws money.
- **Deposit Cash:** Customer deposits cash into their account.
- **Check Balance:** Customer views account balance.
- **Transfer Funds:** Customer transfers funds to another account.
- **Change PIN:** Customer updates their PIN.

Banking System Actions:

- **Authenticate User:** Verifies customer credentials.
- **Process Transactions:** Manages all requests (withdrawals, deposits, transfers).
- **Update Account Data:** Updates balances and transaction history.

Administrator Actions:

- **Monitor Transactions:** Reviews transaction logs for security and performance.
- **Update ATM Settings:** Configures limits, language, and regional settings.

Technician Actions:

- **Load Cash:** Fills the ATM with sufficient cash.
- **Maintain ATM:** Repairs hardware/software and ensures the system operates smoothly.

3. Draw the UML Diagram Using a CASE Tool

- Use **Actors** to represent the **Customer, Banking System, Administrator, and Technician**.

- Use **Use Cases** to represent system functionalities, such as **Withdraw Cash**, **Deposit Cash**, and **Load Cash**.
- Connect **Actors** to their respective **Use Cases** with associations.
- Group related functionalities under the **ATM System** boundary for clarity.

OUTPUT:



RESULT:

The **UML Diagram** for the **ATM System** is successfully created using a CASE tool. It clearly represents the interactions between **Customer**, **Banking System**, **Administrator**, and **Technician** and their respective functionalities, ensuring a comprehensive understanding of the system's workflow.

Hence, the ATM System's processes and actor roles are effectively modeled.