State-Based Testing

1. Introduction

State-based testing was conducted to validate that the banking system enforces correct behavior according to the **account lifecycle state diagram**. The system defines three main operational states for a client account: **Verified**, **Suspended**, and **Closed**. Each state allows or restricts specific actions. The objective of this testing is to ensure that **legal state transitions are permitted** and **illegal transitions are safely rejected**, preserving system integrity and business rules.

---

2. State Transition Matrix

The following table represents the state transition matrix derived directly from the provided state diagram. Each test case defines the current state, triggering action, expected next state, and expected system behavior.

Test ID	Current State	Action	Input Data	Expected Next State	Expected Behavior / Output	Transaction Type
ST01	Verified	Deposit	Amount: $500	Verified	Deposit successful; balance increased by $500; state remains Verified	Legal
ST02	Verified	Withdraw	Amount: $200	Verified	Withdrawal successful; balance decreased by $200; state remains Verified	Legal
ST03	Verified	Violation	Policy breach detected	Suspended	Account suspended due to policy violation; user notified; transactions blocked	Legal
ST04	Verified	AdminAction	Admin closure request	Closed	Account permanently closed by administrator; no further transactions allowed	Legal

Test ID	Current State	Action	Input Data	Expected Next State	Expected Behavior / Output	Transaction Type
ST05	Suspended	View	Account inquiry request	Suspended	Account details displayed; state remains Suspended; read-only access	Legal
ST06	Suspended	Appeal	Successful appeal submission	Verified	Account reinstated to Verified status; normal operations resume	Legal
ST07	Suspended	Withdraw	Amount: $100	Suspended	Error: Transaction denied – account is suspended; balance unchanged	Illegal
ST08	Suspended	Transfer	Amount: $300, Target Account: 12345	Suspended	Error: Transfer denied – account is suspended; balance unchanged	Illegal
ST09	Closed	View	Account inquiry request	Closed	Read-only account details displayed; historical data accessible	Legal
ST10	Closed	Deposit	Amount: $1000	Closed	Error: Deposit denied – account is permanently closed	Illegal
ST11	Closed	Withdraw	Amount: $50	Closed	Error: Withdrawal denied – account is permanently closed	Illegal
ST12	Closed	Appeal	Appeal submission	Closed	Error: Appeal denied – closed accounts cannot be reinstated	Illegal

---

3. Legal State-Based Test Scenarios

Legal state-based test scenarios verify that the system allows only valid transitions defined by the state diagram:

- A **Verified** account can perform financial transactions such as deposit and withdrawal.
- A **Verified → Suspended** transition occurs when a violation is detected.
- A **Suspended → Verified** transition is allowed through a successful appeal.
- An administrator can transition an account from **Verified → Closed**.

- **Suspended** and **Closed** accounts are allowed view-only access.

These tests confirm correct enforcement of business rules and expected state progression.

---

4. Illegal State-Based Test Scenarios

Illegal transitions were tested to ensure the system prevents unauthorized actions:

- Withdrawals and transfers attempted while the account is **Suspended** are rejected.
- Deposits, withdrawals, and appeals attempted on a **Closed** account are denied.
- No transition is allowed from **Closed** to any other state.

In all illegal cases, the system maintains the current state and displays appropriate error messages.

---

5. Test Results Summary

For each state-based test case:

- **Expected outcomes** were clearly defined based on the state diagram.
- **Illegal actions** were correctly blocked.
- **State consistency** was maintained across all transitions.
- No unexpected or undefined transitions were observed.

The results demonstrate that the system correctly implements state-dependent access control and lifecycle constraints.

---

6. Conclusion

State-based testing confirms that the banking system strictly adheres to the defined account lifecycle. Legal transitions are executed correctly, while illegal transitions are safely rejected without affecting account integrity. This testing approach ensures robustness, prevents unauthorized operations, and validates compliance with system requirements defined in the state diagram.