

**Screenshots Portfolio Project Module 8: ATM Operations**

Alexander Ricciardi

Colorado State University Global

CSC505: Principles of Programming

Professor: Dr. Joseph Issa

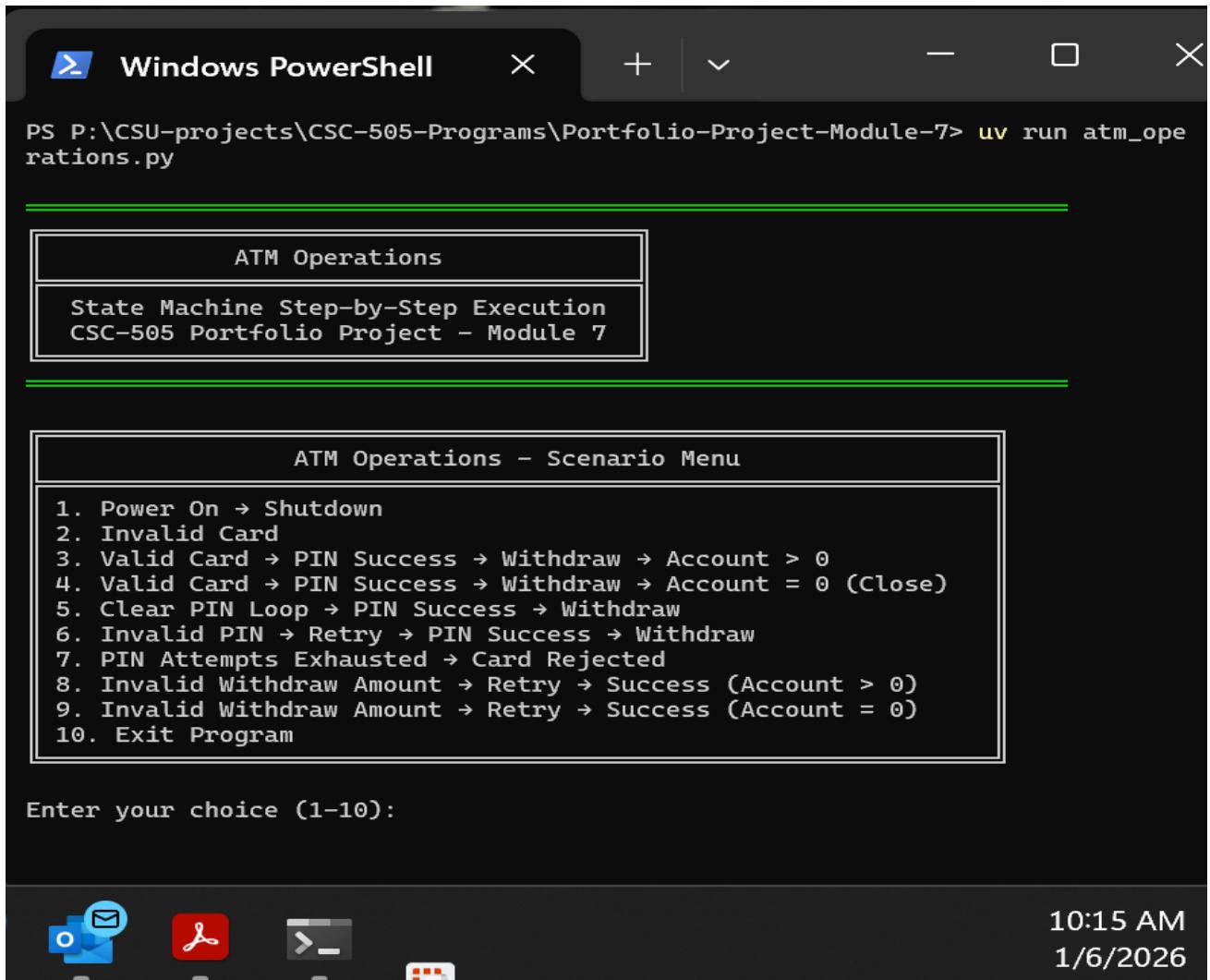
December 11, 2025

### Screenshots Portfolio Project Module 8: ATM Operations

This document contains a screenshot of the ATM Operations script user interactions and outputs

**Figure 1**

Main menu



A screenshot of a Windows PowerShell window titled "Windows PowerShell". The command entered is "PS P:\CSU-projects\CSC-505-Programs\Portfolio-Project-Module-7> uv run atm\_operations.py". Below the command, a green horizontal bar spans the width of the window. Underneath the bar, there are two stacked rectangular boxes. The top box is labeled "ATM Operations" and "State Machine Step-by-Step Execution CSC-505 Portfolio Project - Module 7". The bottom box is labeled "ATM Operations - Scenario Menu" and contains a numbered list from 1 to 10 describing various ATM operations scenarios. At the bottom of the window, the text "Enter your choice (1-10):" is displayed. The taskbar at the bottom of the screen shows icons for Mail, File Explorer, Task View, and Edge, along with the system clock showing 10:15 AM and the date 1/6/2026.

```
PS P:\CSU-projects\CSC-505-Programs\Portfolio-Project-Module-7> uv run atm_operations.py

ATM Operations
State Machine Step-by-Step Execution
CSC-505 Portfolio Project - Module 7

ATM Operations - Scenario Menu
1. Power On → Shutdown
2. Invalid Card
3. Valid Card → PIN Success → Withdraw → Account > 0
4. Valid Card → PIN Success → Withdraw → Account = 0 (Close)
5. Clear PIN Loop → PIN Success → Withdraw
6. Invalid PIN → Retry → PIN Success → Withdraw
7. PIN Attempts Exhausted → Card Rejected
8. Invalid Withdraw Amount → Retry → Success (Account > 0)
9. Invalid Withdraw Amount → Retry → Success (Account = 0)
10. Exit Program

Enter your choice (1-10):
```

10:15 AM  
1/6/2026

See next page.

**Figure 2***Scenario 1 Power On → Shutdown*

```

Windows PowerShell
PS P:\CSU-projects\CSC-505-Programs\Portfolio-Project-Module-7> uv run atm_operations.py

ATM Operations
State Machine Step-by-Step Execution
CSC-505 Portfolio Project - Module 7

ATM Operations - Scenario Menu
1. Power On → Shutdown
2. Invalid Card
3. Valid Card → PIN Success → Withdraw → Account > 0
4. Valid Card → PIN Success → Withdraw → Account = 0 (Close)
5. Clear PIN Loop → PIN Success → Withdraw
6. Invalid PIN → Retry → PIN Success → Withdraw
7. PIN Attempts Exhausted → Card Rejected
8. Invalid Withdraw Amount → Retry → Success (Account > 0)
9. Invalid Withdraw Amount → Retry → Success (Account = 0)
10. Exit Program

Enter your choice (1-10): 1

SCENARIO 1

Scenario: Power On → Shutdown
ATM powers on, displays welcome, then shuts down
Initial Context: account=$100, attempt=0

01 START [*] Initial State
    State machine begins execution from the initial pseudo-state.
02 TRANS PowerOn / showWelcome
    System event: ATM powers on and initializes the welcome display.
03 STATE Idle: entry / showingWelcome
    ATM enters Idle state; system displays welcome screen to attract users.
04 STATE Idle: exit / readCard or shutdown
    User action required: insert card to begin session, or operator shuts down system.
05 TRANS Shutdown / poweringOff
    Operator event: ATM is being shut down for maintenance or end of day.
06 END [*] Final State - System shutdown
    State machine reaches final state. Session outcome: System shutdown.

Scenario Complete
Result: System shutdown
Final Context: account=$100, attempt=0

Press Enter to continue...

```

10:30 AM  
1/6/2026

**Figure 3***Scenario 2 Invalid Card*

```

Windows PowerShell
ATM Operations - Scenario Menu
1. Power On → Shutdown
2. Invalid Card
3. Valid Card → PIN Success → Withdraw → Account > 0
4. Valid Card → PIN Success → Withdraw → Account = 0 (Close)
5. Clear PIN Loop → PIN Success → Withdraw
6. Invalid PIN → Retry → PIN Success → Withdraw
7. PIN Attempts Exhausted → Card Rejected
8. Invalid Withdraw Amount → Retry → Success (Account > 0)
9. Invalid Withdraw Amount → Retry → Success (Account = 0)
10. Exit Program

Enter your choice (1-10): 2

SCENARIO 2

Scenario: Invalid Card
User inserts invalid card → rejected → removes card → session ends
Initial Context: account=$100, attempt=0

01 START [*] Initial State
    State machine begins execution from the initial pseudo-state.
02 TRANS PowerOn / showWelcome
    System event: ATM powers on and initializes the welcome display.
03 STATE Idle: entry / showingWelcome
    ATM enters Idle state; system displays welcome screen to attract users.
04 STATE Idle: exit / readCard or shutdown
    User action required: insert card to begin session, or operator shuts down system.
05 TRANS InsertedCard(validCard) / checkCard
    User action: customer inserts their ATM card into the reader.
06 CHOICE CardCheck: [isValidCard=false] → InvalidCard
    Guard evaluates TRUE: card is invalid. Route to rejection handling.
07 STATE Invalid Card: entry / showingError
    ATM enters Invalid Card state; system shows error message to user.
08 STATE Invalid Card: exit / userAcknowledge && userRemoveCard
    User action required: acknowledge error and remove the rejected card.
09 TRANS UserRemovedCard / endSession
    User removes rejected card; system returns to idle and ends the session.
10 STATE Idle: entry / showingWelcome
    ATM enters Idle state; system displays welcome screen to attract users.
11 STATE Idle: exit / readCard or shutdown
    User action required: insert card to begin session, or operator shuts down system.
12 END [*] Final State - Session ended - invalid card
    State machine reaches final state. Session outcome: Session ended - invalid card.

Scenario Complete
Result: Session ended - card rejected
Final Context: account=$100, attempt=0

Press Enter to continue...

```

10:31 AM  
1/6/2026

**Figure 4**

Scenario 3 Valid Card → PIN Success → Withdraw → Account > 0

```

ATM Operations - Scenario Menu
1. Power On & Shutdown
2. Invalid Card
3. Valid Card & PIN Success & Withdraw & Account > 0
4. Valid Card & PIN Success & Withdraw & Account = 0 (Close)
5. Clean PIN Loop - PIN Success & Withdraw
6. Invalid PIN + Retry & PIN Success & Withdraw
7. PIN Attempts Exhausted & Card Rejected
8. Invalid Withdraw Amount & Retry & Success (Account > 0)
9. Invalid Withdraw Amount & Retry & Success (Account = 0)
10. Exit Program

Enter your choice (1-10): 3

SCENARIO 3

Scenario: Valid Card → PIN Success → Account > 0
Full successful transaction with remaining balance
Initial Context: account=$100, attempt=0

01 START [*] Initial State
    State machine begins execution from the initial pseudo-state.
02 TRANS PowerOn / showWelcome
    System event: ATM powers on and initializes the welcome display.
03 STATE Idle: entry / showingWelcome
    System event: idle state - system displays welcome screen to attract users.
04 STATE Idle: exit / readCard or shutdown
    User action required: insert card to begin session, or operator shuts down system.
05 TRANS InsertedCard(validCard) / checkCard
    User action: customer inserts their ATM card into the reader.
06 CHOICE CardCheck: [isValidCard=true & attempt=0] + PromptPin
    Guard evaluates TRUE: card is valid and no prior attempts. Proceed to PIN entry.
07 STATE Prompt User For Pin: entry / promptingForPin
    ATM enters PIN prompt state; system waits for user to enter their PIN.
08 STATE Prompt User For Pin: exit / pinEntered
    User action required: submit PIN entry to proceed with authentication.
09 TRANS PinEntered(pin, attempt) / incrementAttempt
    User submits PIN; system routes to increment the attempt counter.
10 STATE Increment Attempt: entry / incrementingAttempt
    ATM enters Increment Attempt state; system begins tracking this PIN try.
11 STATE Increment Attempt: do / attempt = attempt + 1 (attempt: 0 + 1)
    System increments attempt counter: 0 + 1.
12 STATE Increment Attempt: exit / attemptIncrementedSuccessfully
    System completed: attempt counter updated successfully.
13 TRANS PinEntered(pin, attempt) / authenticating
    System forwards PIN and attempt count to authentication module.
14 STATE Authenticating: entry / authenticatingPin
    ATM enters Authenticating state; system verifies PIN against bank records.
15 STATE Authenticating: exit / pinValidation
    System completed: PIN validation result determined (valid or invalid).
16 TRANS PinValidation(pin, attempt) / check
    System event: PIN validation result is evaluated.
17 CHOICE PinCheck: [isValidPin=true] + PromptPin
    Guard evaluates TRUE: PIN is correct. Proceed to withdrawal menu.
18 STATE Prompt User For Amount To Withdraw: entry / promptingForWithdrawAmount
    ATM enters Amount Prompt state; system waits for user to enter withdrawal amount.
19 STATE Prompt User For Amount To Withdraw: exit / userEnteredAmount
    User action required: enter the desired withdrawal amount.
20 TRANS Withdraw($50) / computeWithdraw
    User requests $50 withdrawal; system begins balance computation.
21 STATE Compute Withdraw Amount: entry / processingWithdraw
    ATM enters Compute Amount state; system calculates new balance.
22 STATE Compute Withdraw Amount: do / computedAmount = accountAmount - withdrawAmount (100 - 50 = 50)
    System computes: $100 - $50 = $50.
23 STATE Compute Withdraw Amount: exit / withdrawAmountComputed
    System completed: withdrawal amount computation finished.
24 TRANS ProcessAmount(computedAmount) / check
    System event: computed balance is checked for validity.
25 CHOICE AmtCheck: [computedAmount >= 0] + Dispense
    Guard evaluates TRUE: sufficient funds available. Proceed to dispense cash.
26 STATE Dispense Cash: entry / dispensingCash
    ATM enters Dispense Cash state; system prepares to release cash.
27 STATE Dispense Cash: do / accountAmount = computedAmount (accountAmount + 50)
    System updates account balance to $50 after dispensing cash.
28 STATE Dispense Cash: exit / cashDispensed
    System completed: cash has been dispensed to user.
29 TRANS AccountStatus(accountAmount) / check
    System event: account balance is checked after withdrawal.
30 CHOICE AmtCheck: [accountAmount >= 0] + showNewAmt
    Guard evaluates TRUE: balance remains positive. Show new balance.
31 STATE Show Account New Amount: entry / showingAccount
    ATM enters Show Balance state; system displays updated account balance to user.
32 STATE Show Account New Amount: exit / userAcknowledge
    User action required: acknowledge the new balance display.
33 TRANS UserRemovedCard / endSession
    User removes card after viewing balance; session ends successfully.
34 STATE Idle: entry / showingWelcome
    ATM enters Idle state; system displays welcome screen to attract users.
35 STATE Idle: exit / readCard or shutdown
    User action required: insert card to begin session, or operator shuts down system.
36 END [*] Final State - Transaction complete - balance: $50
    State machine reaches final state. Session outcome: Transaction complete - balance: $50.

Scenario Complete
Result: Transaction successful - remaining balance: $50
Final Context: account=$50, attempt=1

Press Enter to continue...

```

**Figure 5***Scenario 4 Valid Card → PIN Success → Withdraw → Account = 0 (Close)*

Windows PowerShell

```

ATM Operations - Scenario Menu
1. Power On → Shutdown
2. Invalid Card
3. Valid Card → PIN Success → Withdraw → Account > 0
4. Valid Card → PIN Success → Withdraw → Account = 0 (Close)
5. Clear PIN Loop → PIN Success → Withdraw
6. Invalid PIN → Retry → PIN Success → Withdraw
7. PIN Attempts Exhausted → Card Rejected
8. Invalid Withdraw Amount → Retry → Success (Account > 0)
9. Invalid Withdraw Amount → Retry → Success (Account = 0)
10. Exit Program

Enter your choice (1-10): 4

SCENARIO 4

Scenario: Valid Card + PIN Success + Account = 0
Full withdrawal depletes account, triggering account closure
Initial Context: account=$100, attempt=0

01 START [*] Initial State
    State machine begins execution from the initial pseudo-state.
02 TRANS PowerOn / showWelcome
    System event: ATM powers on and initializes the welcome display.
03 STATE Idle: entry / showingWelcome
    ATM enters Idle state; system displays welcome screen to attract users.
04 STATE Idle: exit / readCard or shutdown
    User action required: insert card to begin session, or operator shuts down system.
05 TRANS InsertedCard(validCard) / checkCard
    User action: customer inserts their ATM card into the reader.
06 CHOICE CardCheck: [isValidCard=true && attempt==0] → PromptPin
    Guard evaluates TRUE: card is valid and no prior attempts. Proceed to PIN entry.
07 STATE Prompt User For Pin: entry / promptingForPin
    ATM enters PIN prompt state; system waits for user to enter their PIN.
08 STATE Prompt User For Pin: exit / pinEntered
    User action required: submit PIN entry to proceed with authentication.
09 TRANS PinEntered(pin, attempt) / incrementAttempt
    User submits PIN; system routes to increment the attempt counter.
10 STATE Increment Attempt: entry / incrementingAttempt
    ATM enters Increment Attempt state; system begins tracking this PIN try.
11 STATE Increment Attempt: do / attempt = attempt + 1 (attempt: 0 → 1)
    System increments attempt counter: 0 → 1.
12 STATE Increment Attempt: exit / attemptIncrementedSuccessfully
    System completed: attempt counter updated successfully.
13 TRANS PinEntered(pin, attempt) / authenticating
    System forwards PIN and attempt count to authentication module.
14 STATE Authenticating: entry / authenticatingPin
    ATM enters Authenticating state; system verifies PIN against bank records.
15 STATE Authenticating: exit / pinValidation
    System completed: PIN validation result determined (valid or invalid).
16 TRANS PinValidation(isValidPin=true) / check
    System event: PIN validation result is evaluated.
17 CHOICE PinCheck: [isValidPin=true] → PromptAmt
    Guard evaluates TRUE: PIN is correct. Proceed to withdrawal menu.
18 STATE Prompt User For Amount to Withdraw: entry / promptingForWithdrawAmount
    ATM enters Amount Prompt state; system waits for user to enter withdrawal amount.
19 STATE Prompt User For Amount to Withdraw: exit / userEnteredAmount
    User action required: enter the desired withdrawal amount.
20 TRANS Withdraw($100) / computeWithdraw
    User requests full balance ($100); system begins computation.
21 STATE Compute Withdraw Amount: entry / processingWithdraw
    ATM enters Compute Amount state; system calculates new balance.
22 STATE Compute Withdraw Amount: do / computedAmount = accountAmount - withdrawAmount (100 - 100 = 0)
    System computes $100 - $100 = $0.
23 STATE Compute Withdraw Amount: exit / withdrawAmountComputed
    System completed: withdrawal amount computation finished.
24 TRANS ProcessAmount(computedAmount) / check
    System event: computed balance is checked for validity.
25 CHOICE AmtCheck: [computedAmount >= 0] → Dispense
    Guard evaluates TRUE: sufficient funds available. Proceed to dispense cash.
26 STATE Dispense Cash: entry / dispensingCash
    ATM enters Dispense Cash state; system prepares to release cash.
27 STATE Dispense Cash: do / accountAmount = computedAmount (accountAmount ← 0)
    System updates account balance to $0 after dispensing cash.
28 STATE Dispense Cash: exit / cashDispensed
    System completed: cash has been dispensed to user.
29 TRANS AccountStatus(accountAmount) / check
    System event: account balance is checked after withdrawal.
30 CHOICE AcctCheck: [accountAmount == 0] → CloseAccount
    Guard evaluates TRUE: balance is zero. Trigger account closure.
31 STATE Close Account: entry / showingClosureMessage
    ATM enters Close Account state; system processes account closure (balance = $0).
32 STATE Close Account: exit / userAcknowledge && accountClosed
    User action required: acknowledge closure and remove card.
33 TRANS UserRemovedCard / endSession
    User removes card; session ends with account closure processed.
34 STATE Idle: entry / showingWelcome
    ATM enters Idle state; system displays welcome screen to attract users.
35 STATE Idle: exit / readCard or shutdown
    User action required: insert card to begin session, or operator shuts down system.
36 END [*] Final State - Account closed - zero balance
    State machine reaches final state. Session outcome: Account closed - zero balance.

Scenario Complete

Result: Account closed
Final Context: account=$0, attempt=1

Press Enter to continue...

```

10:36 AM  
1/6/2026

**Figure 6****Scenario 5 Clear PIN Loop → PIN Success → Withdraw**

The screenshot shows a Windows PowerShell window titled "Windows PowerShell". Inside, the "ATM Operations - Scenario Menu" is displayed, listing various scenarios from 1 to 10. Scenario 5 is selected, leading to the "SCENARIO 5" screen. This screen details the "Scenario: Clear PIN Loop → Success" and the "Initial Context: account=\$100, attempt=0". The main body of the window shows the step-by-step execution of the scenario, starting with the "START [\*] Initial State" and progressing through various states and transitions until it reaches the final state "Scenario Complete". The log ends with "Result: Transaction successful - remaining balance: \$70" and "Final Context: account=\$70, attempt=1". At the bottom, a message says "Press Enter to continue..." and the system status bar shows the time as 10:38 AM and the date as 1/6/2026.

```

ATM Operations - Scenario Menu
1. Power On > Shutdown
2. Invalid Card
3. Valid Card > PIN Success > Withdraw > Account > 0
4. Valid Card > PIN Success > Withdraw > Account = 0 (Close)
5. Clear PIN Loop > PIN Success > Withdraw
6. Invalid PIN Loop > Retry > PIN Success > Withdraw
7. PIN Attempts Exhausted > Card Rejected
8. Invalid Withdraw Amount > Retry > Success (Account > 0)
9. Invalid Withdraw Amount > Retry > Success (Account = 0)
10. Exit Program

Enter your choice (1-10): 5

SCENARIO 5

Scenario: Clear PIN Loop → Success
User clears PIN entry once, then enters correct PIN
Initial Context: account=$100, attempt=0

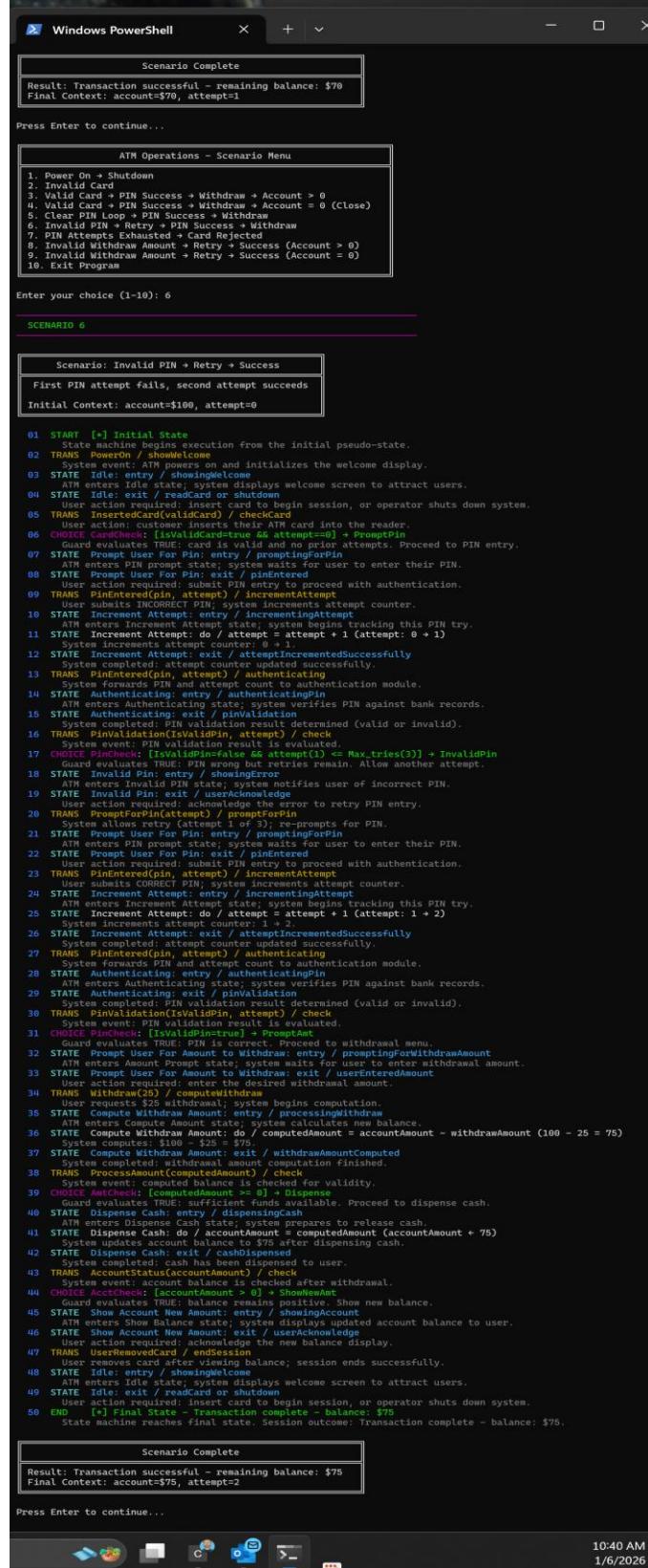
01 START [*] Initial State
    State machine begins execution from the initial pseudo-state.
02 TRANS PowerOn / shutdown
    System event: ATM powers on and initializes the welcome display.
03 STATE Idle: enter / showingWelcome
    ATM enters Idle state; system displays welcome screen to attract users.
04 STATE Idle: exit / readCard or shutdown
    User action required: insert card to begin session, or operator shuts down system.
05 TRANS InsertedCard(validCard) / checkCard
    User action: customer inserts their ATM card into the reader.
06 CHOICE CardCheck: [isValidCard=true && attempt==0] + PromptPin
    Guard evaluates TRUE: card is valid and no prior attempts. Proceed to PIN entry.
07 STATE Prompt User For Pin: entry / promptingForPin
    ATM enters PIN prompt state; system waits for user to enter their PIN.
08 STATE Prompt User For Pin: exit / pinEntered
    User action required: submit PIN entry to proceed with authentication.
09 TRANS UserClearsPin(pin / promptForPin && attempt>0)
    User clears PIN input (e.g., pressed 'Clear' button); system re-prompts for PIN.
10 STATE Prompt User For Pin: entry / promptingForPin
    ATM enters PIN prompt state; system waits for user to enter their PIN.
11 STATE Prompt User For Pin: exit / pinEntered
    User action required: submit PIN entry to proceed with authentication.
12 TRANS PinEntered(pin, attempt) / incrementAttempt
    User submits PIN (after clearing); system increments attempt counter.
13 STATE Increment Attempt: entry / incrementAttempt
    ATM enters Increment Attempt state; system begins tracking this PIN try.
14 STATE Increment Attempt: do / attempt = attempt + 1 (attempt: 0 > 1)
    System increments attempt counter: 0 > 1.
15 STATE Increment Attempt: exit / attemptIncrementedSuccessfully
    System increments attempt counter and succeeds.
16 TRANS PinEntered(pin, attempt) / authenticatePin
    System forwards PIN and attempt count to authentication module.
17 STATE Authenticating: entry / authenticatingPin
    ATM enters Authenticating state; system verifies PIN against bank records.
18 STATE Authenticating: exit / pinValidation
    System completed: PIN validation result determined (valid or invalid).
19 TRANS PinValidation(isValidPin, attempt) / check
    System event: PIN validation result is evaluated.
20 CHOICE PinCheck: [isValidPin=true] + PromptAmt
    Guard evaluates TRUE: PIN is correct. Proceed to withdrawal menu.
21 STATE Prompt User For Amount to Withdraw: entry / promptingForWithdrawAmount
    ATM enters Amount Prompt state; system waits for user to enter withdrawal amount.
22 STATE Prompt User For Amount to Withdraw: exit / userEnteredAmount
    User action required: enter withdrawal amount.
23 TRANS Withdraw(30) / computeWithdraw
    User requests $30 withdrawal; system begins computation.
24 STATE Compute Withdraw Amount: entry / processingWithdraw
    ATM enters Compute Amount state; system calculates new balance.
25 STATE Compute Withdraw Amount: do / computedAmount = accountAmount - withdrawAmount (100 - 30 = 70)
    System computes: $100 - $30 = $70.
26 STATE Compute Withdraw Amount: exit / withdrawAmountComputed
    System completed: withdrawal amount computation finished.
27 TRANS ProcessAmount(computedAmount) / check
    System event: computed balance is checked for validity.
28 CHOICE AmtCheck: [computedAmount >= 0] + Dispense
    Guard evaluates TRUE: sufficient funds available. Proceed to dispense cash.
29 STATE Dispense Cash: entry / dispensingCash
    ATM enters Dispense Cash state; system prepares to release cash.
30 STATE Dispense Cash: do / accountAmount = computedAmount (accountAmount ← 70)
    System updates account balance to $70 after dispensing cash.
31 STATE Dispense Cash: exit / cashDispensed
    System completed: cash has been dispensed to user.
32 TRANS AccountStatus(accountAmount) / check
    System event: account balance is checked after withdrawal.
33 CHOICE AcctCheck: [accountAmount > 0] + ShowNewAmt
    Guard evaluates TRUE: balance remains positive. Show new balance.
34 STATE Show Account New Amount: entry / showingAccount
    ATM enters Show Balance state; system displays updated account balance to user.
35 STATE Show Account New Amount: exit / userAcknowledge
    User action required: acknowledge the new balance display.
36 TRANS ReadCard / doSession
    User removes card after viewing balance; session ends successfully.
37 STATE Idle: enter / showingWelcome
    ATM enters Idle state; system displays welcome screen to attract users.
38 STATE Idle: exit / readCard or shutdown
    User action required: insert card to begin session, or operator shuts down system.
39 END [*] Final State - Transaction complete - balance: $70
    State machine reaches final state. Session outcome: Transaction complete - balance: $70.

Scenario Complete

Result: Transaction successful - remaining balance: $70
Final Context: account=$70, attempt=1

Press Enter to continue...

```

**Figure 7***Scenario 6 Invalid PIN → Retry → PIN Success → Withdraw*

**Figure 8***Scenario 7 PIN Attempts Exhausted → Card Rejected*

Windows PowerShell

```

Scenario Complete
Result: Transaction successful - remaining balance: $75
Final Context: account=$75, attempt=2

Press Enter to continue...

```

ATM Operations - Scenario Menu

1. Power On / Shutdown
2. Invalid Card
3. Valid Card + PIN Success + Withdraw = Account > 0
4. Valid Card + PIN Success + Withdraw = Account = 0 (Close)
5. Clear PIN Loop
6. Invalid PIN + Retry + PIN Success + Withdraw
7. ATM Attempts Exceeded
8. Invalid Withdraw Amount + Retry + Success (Account > 0)
9. Invalid Withdraw Amount + Retry + Success (Account = 0)
10. Exit Program

Enter your choice (1-10): 7

SCENARIO 7

Scenario: PIN Attempts Exhausted

User fails PIN 4 times, card is rejected

Initial Context: account=\$100, attempt=0

```

01 START [*] Initial State
    STATE machine begins execution from the initial pseudo-state.
02 TRANS ShowWelcome
    System event: ATM powers on and initializes the welcome display.
03 STATE Idle: entry / showingWelcome
    User action required: insert card into the reader.
04 STATE Idle: exit / shutdown
    User action required: insert card into the reader.
05 TRANS InsertedCard(validCard) / checkCard
    Guard evaluates TRUE: card is valid and no prior attempts. Proceed to PIN entry.
06 CHOICE CardCheck: [isValidCard=true && attempt=0] + PromptPin
    Guard evaluates TRUE: card is valid and no prior attempts. Proceed to PIN entry.
07 STATE Prompt User For Pin: entry / promptingForPin
    User action required: enter PIN for user to enter their PIN.
08 STATE Prompt User For Pin: exit / pinEntered
    User action required: submit PIN entry to proceed with authentication.
09 TRANS PinEntered(pin) / incrementAttempt
    User action required: submit PIN entry to proceed with authentication.
    System increments attempt counter by 1; increments counter.
10 STATE Increment: attempt / incrementingAttempt
    ATM enters Increment Attempt state; system begins tracking this PIN try.
11 STATE Increment: attempt / attemptIncrementedSuccessfully
    System increments attempt counter: 0 + 1.
12 STATE Increment: attempt / attemptIncrementedSuccessfully
    System completed: attempt incremented successfully.
13 TRANS PinEntered(pin, attempt) / authenticating
    System forwards PIN and attempt count to authentication module.
14 STATE Authenticating: entry / authenticatingPin
    ATM enters Authenticating state; system verifies PIN against bank records.
15 STATE Authenticating: exit / pinValidation
    System completed: PIN validation result determined (Valid or invalid).
16 TRANS PinValidation(isValidPin, attempt) / check
    Guard evaluates TRUE: PIN validation result determined (Valid or invalid).
17 CHOICE PinCheck: [isValidPin=false && attempt(1) == Max_tries(3)] + InvalidPin
    Guard evaluates TRUE: PIN wrong and no retries remain. Allow another attempt.
18 STATE Increment: attempt / attemptIncrementedSuccessfully
    System increments attempt counter: 1 + 1.
19 STATE Increment: attempt / attemptIncrementedSuccessfully
    System completed: attempt updated successfully.
20 STATE Invalid Pin: entry / userAcknowledge
    User action required: acknowledge the error to retry PIN entry.
21 TRANS PromptUserForPin: entry / promptingForPin
    System allows retry (attempt 2 + 3); re-prompts for PIN.
22 STATE Prompt User For Pin: entry / pinEntered
    ATM enters PIN prompt state; system waits for user to enter their PIN.
23 STATE Prompt User For Pin: exit / pinEntered
    User action required: submit PIN entry to proceed with authentication.
24 TRANS PinEntered(pin, attempt) / incrementAttempt
    User submits INCORRECT PIN (attempt #2); system increments counter.
25 STATE Increment: attempt / attemptIncrementedSuccessfully
    ATM enters Increment Attempt state; system begins tracking this PIN try.
26 STATE Increment: attempt / attemptIncrementedSuccessfully
    System increments attempt counter: 1 + 2.
27 STATE Increment: attempt / attemptIncrementedSuccessfully
    System completed: attempt updated successfully.
28 STATE PinEntered(pin, attempt) / authenticating
    System forwards PIN and attempt count to authentication module.
29 STATE Authenticating: entry / authenticatingPin
    ATM enters Authenticating state; system verifies PIN against bank records.
30 STATE Authenticating: exit / pinValidation
    System completed: PIN validation result determined (Valid or invalid).
31 TRANS PinValidation(isValidPin, attempt) / check
    System event: PIN validation result is evaluated.
32 CHOICE PinCheck: [isValidPin=false && attempt(2) == Max_tries(3)] + InvalidPin
    Guard evaluates TRUE: PIN wrong and no retries remain. Allow another attempt.
33 STATE Invalid Pin: entry / showingError
    ATM enters Invalid PIN state; system notifies user of incorrect PIN.
34 STATE Invalid Pin: entry / userAcknowledge
    User action required: acknowledge the error to retry PIN entry.
35 TRANS PromptForPin(attempt) / promptForPin
    System allows retry (attempt 2 + 3); re-prompts for PIN.
36 STATE Prompt User For Pin: entry / promptingForPin
    ATM enters PIN prompt state; system waits for user to enter their PIN.
37 STATE Prompt User For Pin: exit / pinEntered
    User action required: submit PIN entry to proceed with authentication.
38 TRANS PinEntered(pin, attempt) / incrementAttempt
    User submits INCORRECT PIN (attempt #3); system increments counter.
39 STATE Increment: attempt / incrementingAttempt
    ATM enters Increment Attempt state; system begins tracking this PIN try.
40 STATE Increment: attempt / attemptIncrementedSuccessfully
    System increments attempt counter: 2 + 1.
41 STATE Increment: attempt / attemptIncrementedSuccessfully
    System completed: attempt updated successfully.
42 STATE PinEntered(pin, attempt) / authenticating
    System forwards PIN and attempt count to authentication module.
43 STATE Authenticating: entry / authenticatingPin
    ATM enters Authenticating state; system verifies PIN against bank records.
44 STATE Authenticating: exit / pinValidation
    System completed: PIN validation result determined (Valid or invalid).
45 TRANS PinValidation(isValidPin, attempt) / check
    Guard evaluates TRUE: PIN validation result is evaluated.
46 CHOICE PinCheck: [isValidPin=false && attempt(3) == Max_tries(3)] + InvalidPin
    Guard evaluates TRUE: PIN wrong and no retries remain. Allow another attempt.
47 STATE Invalid Pin: entry / userAcknowledge
    ATM enters Invalid PIN state; system notifies user of incorrect PIN.
48 STATE Invalid Pin: exit / userAcknowledge
    User action required: acknowledge the error to retry PIN entry.
49 TRANS PromptForPin(attempt) / promptForPin
    System allows retry (attempt 3 + 3); re-prompts for PIN.
50 STATE PinEntered(pin) / incrementAttempt
    ATM enters PIN prompt state; system waits for user to enter their PIN.
51 STATE Prompt User For Pin: entry / pinEntered
    User action required: submit PIN entry to proceed with authentication.
52 TRANS PinEntered(pin, attempt) / incrementAttempt
    User submits INCORRECT PIN (attempt #4); system increments counter.
53 STATE Increment: attempt / incrementingAttempt
    ATM enters Increment Attempt state; system begins tracking this PIN try.
54 STATE Increment: attempt / attemptIncrementedSuccessfully
    System increments attempt counter: 3 + 1.
55 STATE Increment: attempt / attemptIncrementedSuccessfully
    System completed: attempt updated successfully.
56 STATE PinEntered(pin, attempt) / authenticating
    System forwards PIN and attempt count to authentication module.
57 STATE Authenticating: entry / authenticatingPin
    ATM enters Authenticating state; system verifies PIN against bank records.
58 STATE Authenticating: exit / pinValidation
    System completed: PIN validation result determined (Valid or invalid).
59 TRANS PinValidation(isValidPin, attempt) / check
    Guard evaluates TRUE: PIN validation result is evaluated.
60 CHOICE PinCheck: [isValidPin=false && attempt(4) > Max_tries(3)] + AttemptExhausted
    Guard evaluates TRUE: PIN wrong and no retries left. Reject card.
61 STATE Attempt Exhausted: entry / userAcknowledge && userRemoveCard
    User action required: acknowledge error and remove rejected card.
62 TRANS UserRemoveCard: entry / userRemoveCard
    User removes rejected card; session ends with security lockout.
63 STATE Idle: entry / showingWelcome

```

64 STATE Idle: exit / readcard or shutdown
 User action required: insert card to begin session, or operator shuts down system.

65 END [\*] Final State - Card rejected - PIN attempts exhausted
 State machine reaches final state. Session outcome: Card rejected - PIN attempts exhausted.

Scenario Complete

Result: Card rejected - too many PIN attempts

Final Context: account=\$100, attempt=4

Press Enter to continue...

10:43 AM  
1/6/2026

**Figure 9***Scenario 8 Invalid Withdraw Amount → Retry → Success (Account > 0)*

```

Windows PowerShell

Scenario Complete
Result: Card rejected - too many PIN attempts
Final Context: account=$100, attempt=4

Press Enter to continue...

ATM Operations - Scenario Menu
1. Power On + Shutdown
2. Invalid Card
3. Valid Card + PIN Success + Withdraw + Account > 0
4. Valid Card + PIN Success + Withdraw + Account = 0 (Close)
5. Clear PIN + PIN Success + Withdraw
6. Invalid PIN + Retry + PIN Success + Withdraw
7. PIN Attempts Exhausted + Card Rejected
8. Invalid Withdraw Amount + Retry + Success (Account > 0)
9. Invalid Withdraw Amount + Retry + Success (Account = 0)
10. Exit Program

Enter your choice (1-10): 8

SCENARIO 8

Scenario: Invalid Amount → Success (Account > 0)
User enters amount exceeding balance, retries with valid amount
Initial Context: account=$100, attempt=0

01 START [*] Initial State
  State machine begins execution from the initial pseudo-state.
02 TRANS PowerOn / showWelcome
  System event: ATM powers on and initializes the welcome display.
03 STATE Idle: entry / showingWelcome
  ATM displays welcome screen to attract users.
04 STATE Idle: exit / readCard or shutdown
  User action required: insert card to begin session, or operator shuts down system.
05 TRANS InsertedCard(validCard) / checkCard
  User action: customer inserts their ATM card into the reader.
06 CHOICE PromptUserForPin [card is valid and no prior attempts] + PromptPin
  Guard evaluates TRUE: card is valid and no prior attempts. Proceed to PIN entry.
07 STATE Prompt User For Pin: entry / promptingForPin
  ATM enters PIN prompt state; system waits for user to enter their PIN.
08 STATE Prompt User For Pin: exit / pinEntered
  User action required: enter PIN, PIN must be processed with authentication.
09 TRANS PinEntered(pin, attempt) / incrementAttempt
  User submits PIN; system increments attempt counter.
10 STATE Increment Attempt: entry / incrementingAttempt
  System event: PIN entered, system begins tracking this PIN try.
11 STATE Increment Attempt: do / attempt + 1 (attempt: 0 + 1)
  System increments attempt counter: 0 + 1.
12 STATE Increment Attempt: exit / attemptIncrementsSuccessfully
  System completes attempt and re-evaluates successfully.
13 TRANS PinEntered(pin, attempt) / authenticating
  System forwards PIN and attempt count to authentication module.
14 STATE Authenticating: entry / authenticatingPin
  ATM enters Authenticating state; system verifies PIN against bank records.
15 STATE Authenticating: exit / pinValidation
  System event: PIN validation result determined (valid or invalid).
16 TRANS PinValidation(isValidPin, attempt) / check
  System event: PIN validation result is evaluated.
17 CHOICE PinCheck: [isValidPin=true] + PromptAmt
  Guard evaluates TRUE: PIN is valid. Proceed to withdrawal menu.
18 STATE Prompt User For Amount to Withdraw: entry / promptingForWithdrawAmount
  ATM enters Amount Prompt state; system waits for user to enter withdrawal amount.
19 STATE Prompt User For Amount to Withdraw: exit / userEnteredAmount
  User enters desired amount, system waits for user to enter withdrawal amount.
20 TRANS Withdraw($100) / computeWithraw
  User requests $100 (EXCEEDS $100 balance); computation begins.
21 STATE Compute Withdraw Amount: entry / processingWithdraw
  ATM enters Compute Amount state; system calculates new balance.
22 STATE Compute Withdraw Amount: do / computedAmount = accountAmount - withdrawAmount (100 - 100 = -50)
  System computes: $100 - $100 = -$50
23 STATE Compute Withdraw Amount: exit / withdrawAmountComputed
  System completed: withdrawal amount computation finished.
24 TRANS ProcessAmount(computedAmount) / check
  System event: computed amount is checked for validity.
25 CHOICE AmtCheck: [computedAmount<-50] < 0] * InvalidAmount
  Guard evaluates TRUE: insufficient funds. Route to error handling.
26 STATE Invalid Amount: entry / showingError
  ATM enters Invalid Amount state; system shows insufficient funds error.
27 STATE Invalid Amount: exit / userAcknowledged
  User action required: acknowledge error to enter a new amount.
28 TRANS InvalidAmount[withdrawAmount] / promptUserForNewAmount
  Insufficient funds selected; system prompts user to enter a smaller amount.
29 STATE Prompt User For New Amount: entry / enteringWithdrawAmount
  ATM enters Amount Prompt state; system waits for user to enter withdrawal amount.
30 STATE Prompt User For New Amount: exit / userEnteredAmount
  User action required: enter the desired withdrawal amount.
31 TRANS Withdraw(40) / computeWithraw
  User enters $40 (valid amount); system begins computation.
32 STATE Compute Withdraw Amount: entry / processingWithdraw
  ATM enters Compute Amount state; system calculates new balance.
33 STATE Compute Withdraw Amount: do / computedAmount = accountAmount - withdrawAmount (100 - 40 = 60)
  System computes: $100 - $40 = $60
34 STATE Compute Withdraw Amount: exit / withdrawAmountComputed
  System completed: withdrawal amount computation finished.
35 TRANS ProcessAmount(computedAmount) / check
  System event: computed balance is checked for validity.
36 CHOICE AmtCheck: [computedAmount>60] * BalanceExceeded
  Guard evaluates TRUE: insufficient funds available. Proceed to dispense cash.
37 STATE Dispense Cash: entry / dispensingCash
  ATM enters Dispense Cash state; system prepares to release cash.
38 STATE Dispense Cash: do / accountBalance = accountAmount + 60
  System updates account balance to $60 after dispensing cash.
39 STATE Dispense Cash: exit / cashDispensed
  System completed: cash has been dispensed to user.
40 TRANS AccountStatus(accountAmount) / check
  System event: account status is checked after withdrawal.
41 CHOICE AcctCheck: [accountAmount > 0] * ShowNewWamt
  Guard evaluates TRUE: balance remains positive. Show new balance.
42 STATE Show Account New Amount: entry / showingAccount
  ATM enters Show Balance state; system displays updated account balance to user.
43 STATE Show Account New Amount: exit / userAcknowledged
  User action required: acknowledge the new balance display.
44 TRANS UserRemovedCard / endSession
  User removes card after viewing balance; session ends successfully.
45 STATE Idle: entry / showingWelcome
  ATM enters Idle state; system displays welcome screen to attract users.
46 STATE Idle: exit / readCard or shutdown
  User action required: insert card to begin session, or operator shuts down system.
47 END [*] Final State - Transaction complete - balance: $60
  State machine reaches final state. Session outcome: Transaction complete - balance: $60.

Scenario Complete
Result: Transaction successful - remaining balance: $60
Final Context: account=$60, attempt=3

Press Enter to continue...

```

**Figure 10***Scenario 9 Invalid Withdraw Amount → Retry → Success (Account = 0)*

```

Scenario Complete
Result: Transaction successful - remaining balance: $60
Final Context: account=$60, attempt=1

Press Enter to continue...

ATM Operations - Scenario Menu
1. Power On > Shutdown
2. Invalid Card
3. Invalid Card + PIN Success > Withdraw + Account > 0
4. Valid Card + PIN Success > Withdraw + Account = 0 (Close)
5. Clear PIN Loop > PIN Success > Withdraw
6. Invalid PIN > Retry + PIN Success > Withdraw
7. Invalid PIN > Invalid Card Rejected
8. Invalid Withdraw Amount > SetPinSuccess (Account > 0)
9. Invalid Withdraw Amount > Retry + Success (Account = 0)
10. Exit Program

Enter your choice (1-10): 9

SCENARIO 9

Scenario: Invalid Amount → Success (Account = 0)
User retries with full balance withdrawal, triggering account closure
Initial Context: account=$100, attempt=0

01 START [*] Initial State
02 TRANS PowerOn / showWelcome
    System event: ATM powers on and initializes the welcome display.
03 STATE Idle: entry / showingWelcome
    ATM enters Idle state; system displays welcome screen to attract users.
04 STATE Idle: exit / readCard or shutdown
    User action required: insert card and begin session, or operator shuts down system.
05 TRANS InsertedCard(validCard) / checkCard
    User action: customer inserts their ATM card into the reader.
06 CHOICE CardCheck: [validCard & attempt < 1] * PromptPin
    Guard evaluates TRUE: card is valid and no prior attempts. Proceed to PIN entry.
07 STATE Prompt User For Pin: entry / promptingForPin
    ATM enters PIN prompt state; system waits for user to enter their PIN.
08 STATE Prompt User For Pin: exit / pinEntered
    ATM exits PIN prompt state; system waits for user to enter their PIN.
09 STATE PinEntered(pin, attempt) / incrementAttempt
    User submits PIN; system increments attempt counter.
10 STATE Increment Attempt: entry / incrementingAttempt
    System event: user increments attempt counter.
11 STATE Increment Attempt: do / attempt < attempt + 1 (attempt: 0 > 1)
    System increments attempt counter: 0 + 1.
12 STATE Increment Attempt: exit / attemptIncrementedSuccessfully
    System event: attempt was successfully incremented.
13 TRANS PinEntered(pin, attempt) / authenticating
    System forwards PIN and attempt count to authentication module.
14 STATE Authenticating: entry / authenticatingPin
    ATM enters Authenticating state; system verifies PIN against bank records.
15 STATE Authenticating: exit / validatePin
    System event: PIN validation result determined (valid or invalid).
16 TRANS PinValidation(isValidPin, attempt) / check
    System event: PIN validation result is evaluated.
17 CHOICE PinValidation: [isValidPin & attempt < 3]
    Guard evaluates TRUE: PIN is correct. Proceed to withdrawal menu.
18 STATE Prompt User For Amount To Withdraw: entry / promptingForWithdrawAmount
    ATM enters Amount Prompt state; system waits for user to enter withdrawal amount.
19 STATE Prompt User For Amount To Withdraw: exit / userEnteredAmount
    User action required: enter the desired withdrawal amount.
20 TRANS Withdraw($200) / computeWithdraw
    User requests $200 (EXCEEDS $100 balance); computation begins.
21 STATE Compute Withdraw Amount: entry / processingWithdraw
    ATM enters Compute Withdraw Amount state; system calculates new balance.
22 STATE Compute Withdraw Amount: do / computedAmount = accountAmount - withdrawAmount (100 - 200 = -100)
    System computes: $100 - $200 = -$100.
23 STATE Compute Withdraw Amount: exit / withdrawAmountComputed
    ATM exits Compute Withdraw Amount state; computation finished.
24 TRANS ProcessAmount(computedAmount) / check
    System event: computed balance is checked for validity.
25 CHOICE AmtCheck: [computedAmount<0] * InvalidAmount
    Guard evaluates TRUE: insufficient funds. Route to error handling.
26 STATE Invalid Amount: entry / userAcknowledge
    ATM enters Invalid Amount state; system shows insufficient funds error.
27 STATE Invalid Amount: exit / userAcknowledge
    User action required: acknowledge error to enter a new amount.
28 TRANS ProcessAmount(computedAmount) / processForWithdraw
    Insufficient funds detector system prompts user to enter a valid amount.
29 STATE Prompt User For Amount To Withdraw: entry / promptingForWithdrawAmount
    ATM enters Amount Prompt state; system waits for user to enter withdrawal amount.
30 STATE Prompt User For Amount To Withdraw: exit / userEnteredAmount
    User action required: enter the desired withdrawal amount.
31 TRANS Withdraw($100) / computeWithdraw
    User requests full balance ($100); system begins computation.
32 STATE Compute Withdraw Amount: entry / processingWithdraw
    ATM enters Compute Withdraw Amount state; system calculates new balance.
33 STATE Compute Withdraw Amount: do / computedAmount = accountAmount - withdrawAmount (100 - 100 = 0)
    System computes: $100 - $100 = $0.
34 STATE Compute Withdraw Amount: exit / withdrawAmountComputed
    ATM exits Compute Withdraw Amount state; computation finished.
35 TRANS ProcessAmount(computedAmount) / check
    System event: computed balance is checked for validity.
36 CHOICE AmtCheck: [computedAmount == 0] * Dispense
    Guard evaluates TRUE: balance is zero. Trigger account closure.
37 STATE Close Account: entry / showingClosureMessage
    ATM enters Close Account state; system processes account closure (balance = $0).
38 STATE Close Account: exit / userAcknowledge && accountClosed
    User action required: acknowledge closure and remove card.
39 TRANS UserRemovedCard / endSession
    User removes card; session ends with account closure processed.
40 STATE Idle: entry / showingWelcome
    ATM enters Idle state; system displays welcome screen to attract users.
41 STATE Idle: exit / readCard or shutdown
    User action required: insert card to begin session, or operator shuts down system.
42 END [*] Final State - Account Closed - zero balance
    System machine reaches final state. Session outcome: Account closed - zero balance.

Scenario Complete
Result: Account closed..
Final Context: account=$0, attempt=1

Press Enter to continue...

```

**Figure 11***Exit Program*