ATM Simulator System

A robust, console-based banking simulation built with Python.

Project Overview

What is it?

The ATM Simulator is a Python-based application that mimics the core functionalities of a real-world Automated Teller Machine (ATM).

Goal

To provide a user-friendly interface for performing basic financial transactions like withdrawals, deposits, and PIN management without the need for physical hardware.

Core Features

Secure Sign-In

Robust authentication system requiring a valid Username and 4-digit PIN.

Statement

View real-time account balance and a history of recent transactions.

Transactions

Deposit (Lodge) and Withdraw cash with instant balance updates.

PIN Management

Security feature allowing users to update their access PIN securely.

Console Interface

Lightweight, text-based menu system for fast and efficient navigation.

Validation

Prevents overdrafts and ensures positive numeric inputs for all actions.

Technical Stack

Programming Language

Built entirely in **Python**, utilizing standard libraries (`sys`) for system operations.

Focuses on core programming concepts like loops, conditionals, and functions.

Data Structure

Uses a **Python Dictionary** (`ACCOUNT_DATA`) to store user information.

- In-memory storage (RAM)
- Fast access time (O(1))
- Stores nested lists for transactions

Authentication Logic

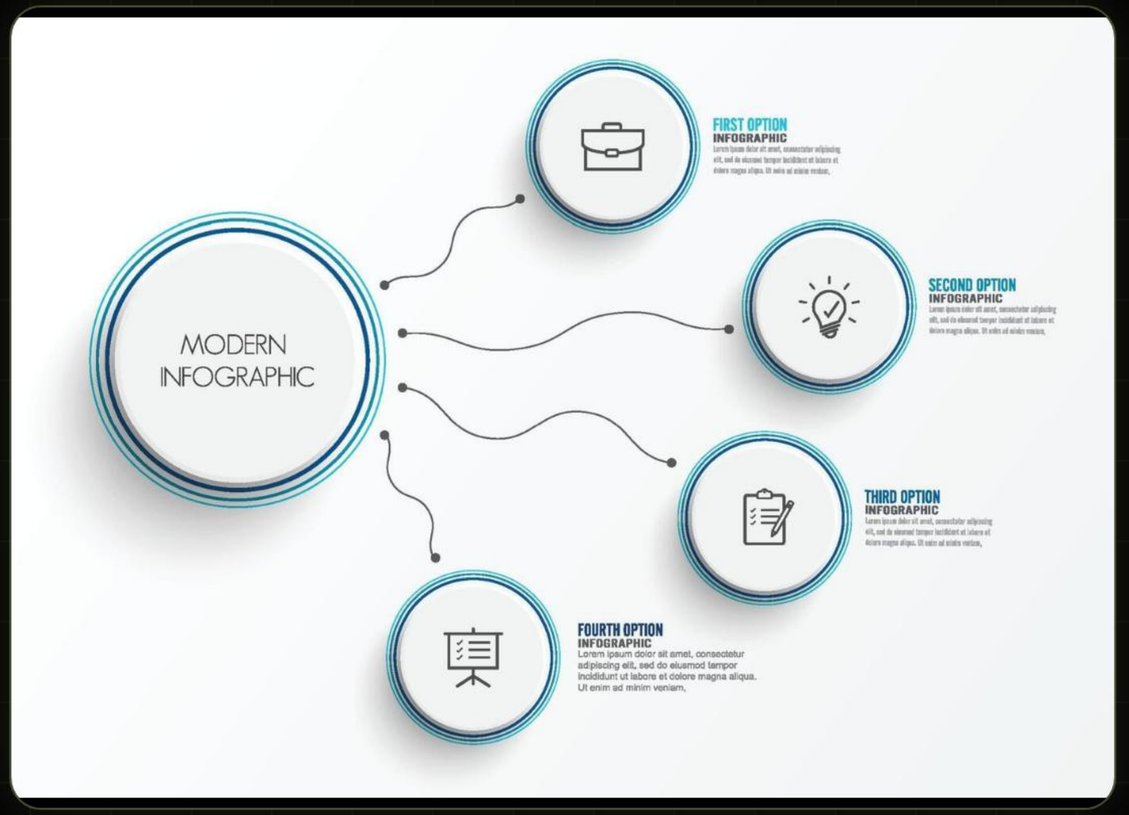
The Sign-In Process

- **Hardcoded Credentials:** The system validates against a stored username ("user123") and PIN ("1234").
- **Attempt Limiter:** A `for` loop grants the user strictly 3 attempts to enter correct details.
- **Security Lockout:** If all 3 attempts fail, `sys.exit()` is called to terminate the program immediately, mimicking a card capture or lockout.

Transaction Algorithm

Withdrawal Logic

1. **Input:** User enters amount.
2. **Validation A:** Is amount numeric and positive (>0)?
3. **Validation B:** Is amount <= Current Balance? (Overdraft Protection)
4. **Execution:** Subtract from Balance -> Append to Transaction Log -> Display Success.

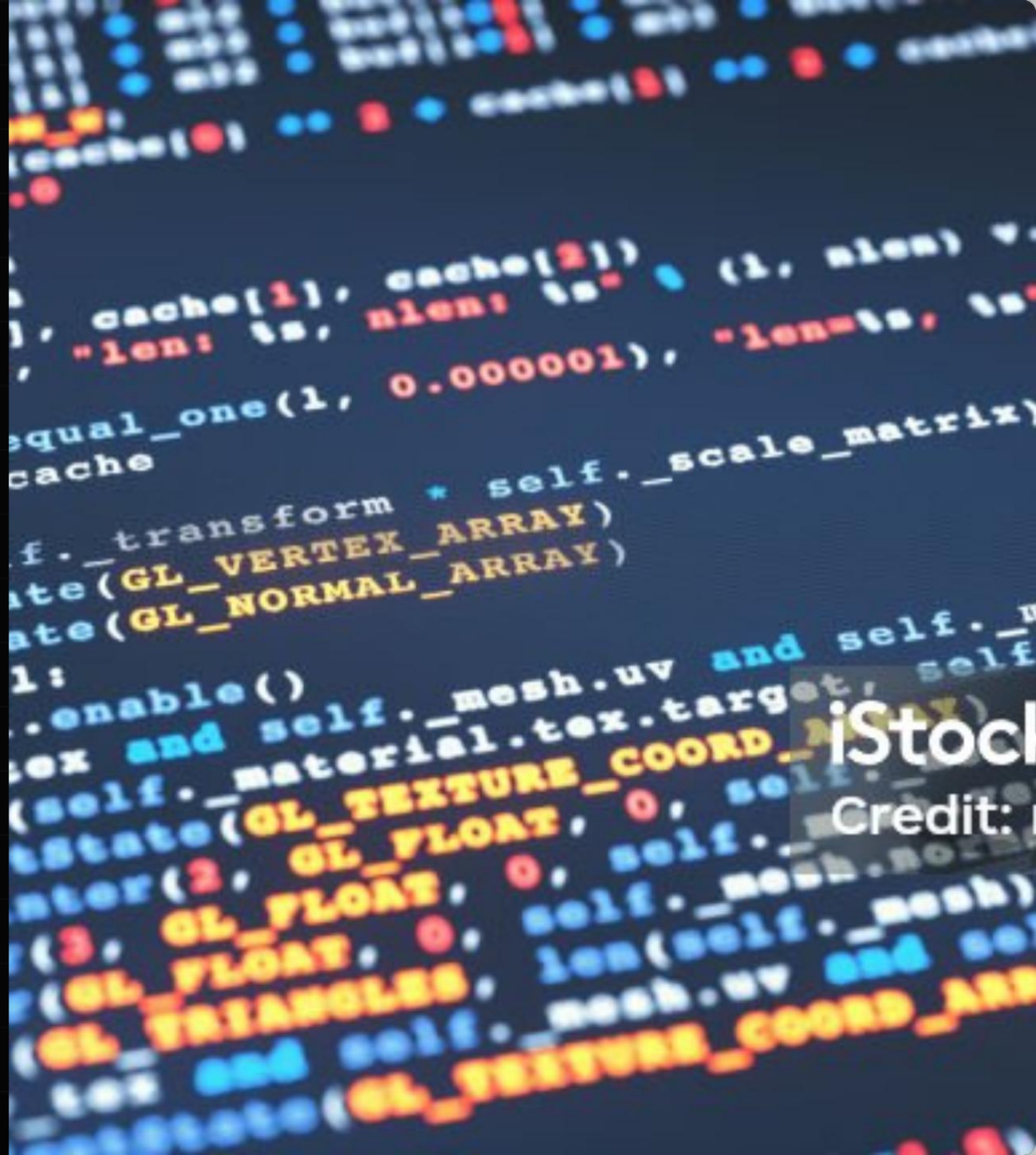


Code Structure

Modular Design

The application is broken down into distinct functions for maintainability and readability:

- **def sign_in():** Handles auth loop.
- **def withdraw_amount():** Logic for debits.
- **def display_statement():** Formats and prints the log.
- **def main():** The central `while True` loop that keeps the menu running until exit.



Simulated Transaction History

A visual representation of how the system tracks debits vs credits in the simulated account.

Future Enhancements

GUI Integration

Replace the console output with a graphical interface using **Tkinter** or **PyQt** for a modern look.

Database

Implement **SQLite** or **MySQL** to persist user data, allowing balances to be saved even after the program closes.

Multi-User Support

Expand the dictionary structure to handle multiple unique usernames and PINs simultaneously.

Conclusion

The ATM Simulator successfully demonstrates the core logic of banking systems using fundamental Python concepts. It serves as an excellent foundation for understanding transaction management, input validation, and secure authentication flows.

Q & A

Thank you for your attention.

Image Sources

https://static.vecteezy.com/system/resources/previews/059/543/458/non_2x/3d-render-illustration-of-a-minimalist-cash-machine-or-atm-icon-that-dispenses-banknotes-financial-concepts-transactions-and-withdrawals-banking-electronic-services-digital-payment-on-white-vector.jpg

Source: www.vecteezy.com

https://img.freepik.com/premium-photo/cyber-security-data-protection-concept-with-digital-lock-circuit-background_1363766-154.jpg

Source: www.freepik.com

https://static.vecteezy.com/system/resources/previews/007/202/652/non_2x/abstract-elements-of-graph-infographic-template-with-label-integrated-circles-business-concept-with-4-options-for-content-diagram-flowchart-steps-parts-timeline-infographics-workflow-layout-vector.jpg

Source: www.vecteezy.com

https://media.istockphoto.com/id/1411610324/sv/foto/python-programming-language-computer-source-code-text-example-close-up-on-a-blue-surface.jpg?s=1024x1024&w=is&k=20&c=oT6axbQLcy-V65ha6CSWtXPI3pIfPOUb4qFBi83nxwo=

Source: www.istockphoto.com