ATM SIMULATION

Vu Hoang Phuc

20214923

OUTLINE

I Problem Introduction

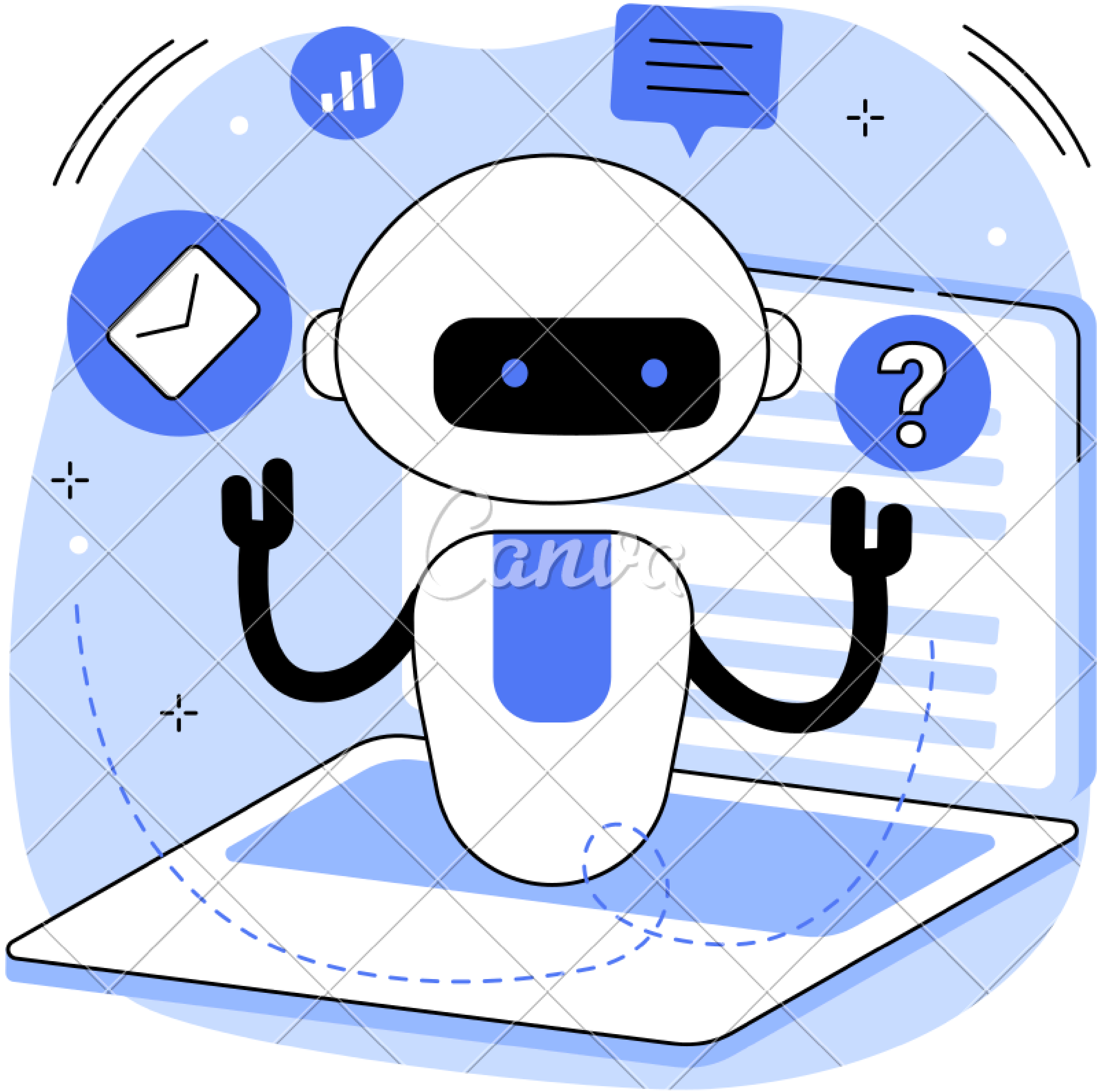
II Components

III Implementation

IV Results

V Conclusion

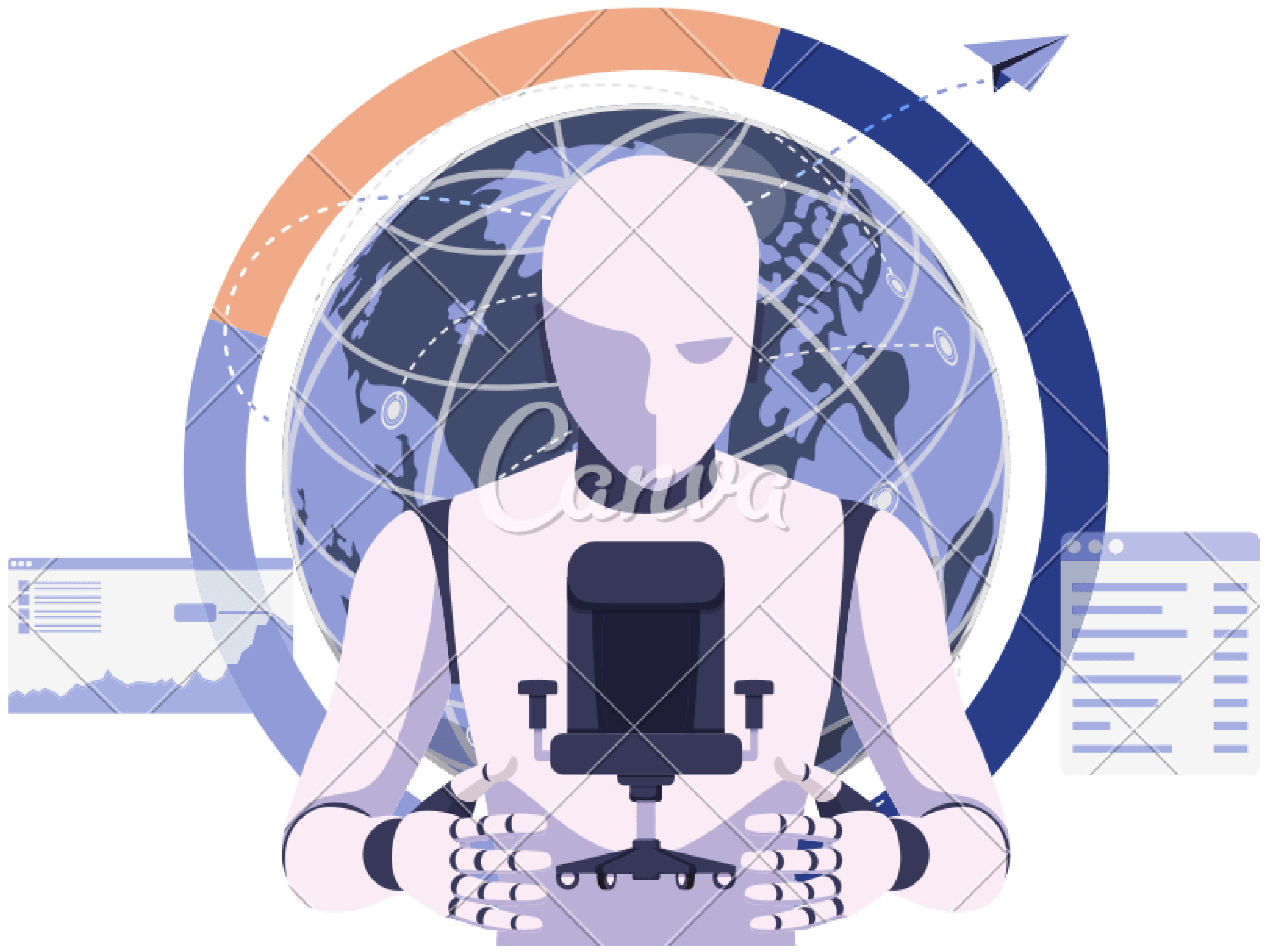
I. PROBLEM INTRODUCTION



BACKGROUND

- The ATM Simulator replicates core banking functionalities to educate users about account management, transaction limits, and financial discipline.
- Uses modern programming techniques and data structures like dictionaries, deques, and heaps to simulate real-world banking scenarios efficiently.
- Enhances financial literacy by providing a safe, interactive environment for practicing transactions and understanding banking algorithms.

II. COMPONENTS



Account Class

- The Account class is fundamental to the simulator, encapsulating the attributes and behaviors associated with a bank account.
- Each instance of the Account class represents a user’s account and includes properties such as account ID, balance, and transaction limits for withdrawals and deposits and their limits.
- Methods within the class, such as withdraw, deposit, and get_transaction_history, enable users to perform transactions while maintaining a history of their activities. The use of a deque for transaction history ensures that users can efficiently access their most recent transactions

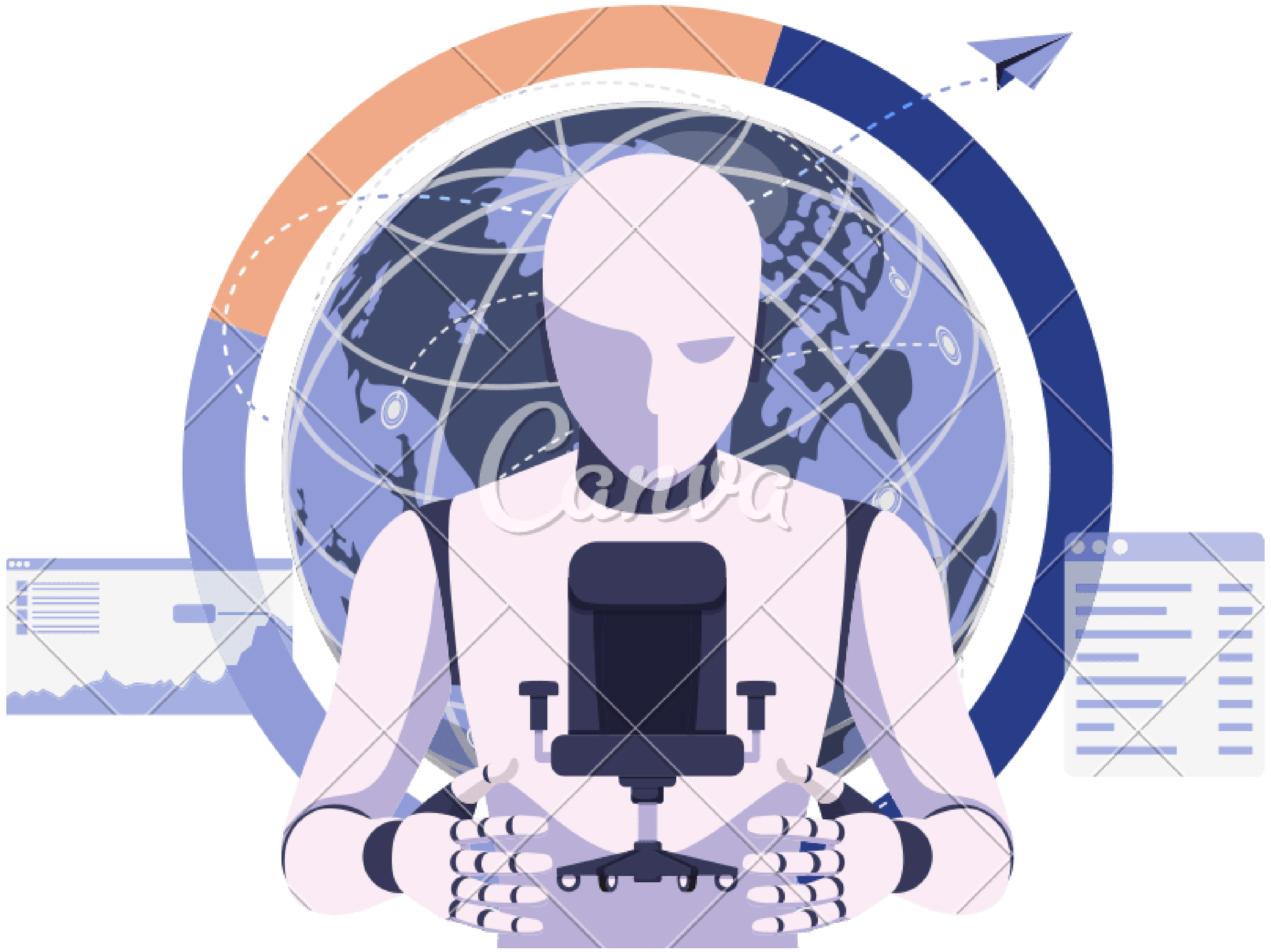
Bank Class

- The Bank class serves as the central management system for handling multiple accounts.
- Maintains a dictionary of accounts, allowing for quick retrieval and management of user data.
- Includes methods for creating new accounts, loading and saving account data to JSON files, and processing daily resets of transaction limits using a min-heap.
- This heap efficiently schedules daily reset events, ensuring that transaction limits are refreshed at the right time.
- Enforces cooldown periods between transactions, preventing users from executing rapid-fire operations and thus mimicking the realistic constraints of banking systems.

User Interaction

- The user interaction is facilitated through a console-based menu system in the main function.
- Users are prompted to enter their account ID (acting as a PIN), which the Bank class uses to retrieve the corresponding Account instance.
- The menu provides options for checking balance, withdrawing, depositing, viewing transaction history, or quitting the application.
- This interactive approach allows users to simulate real banking operations while the underlying code manages the interactions between the user, the Bank, and the Account objects seamlessly.

III. IMPLEMENTATION



Dictionary

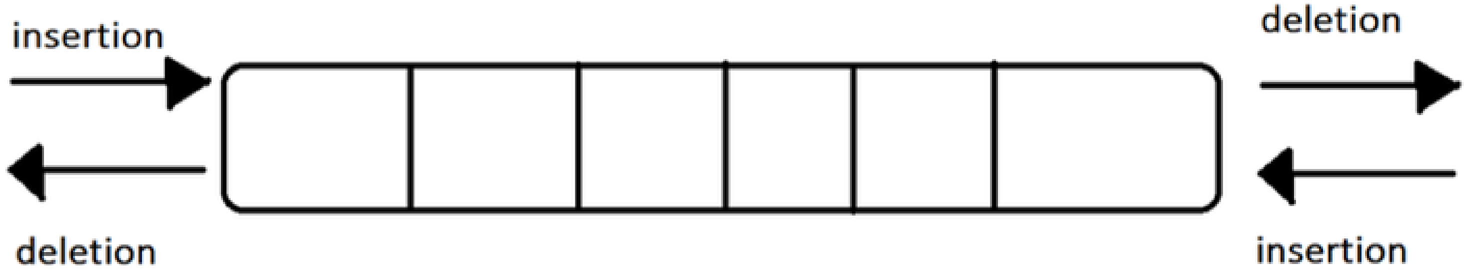
The Bank class utilizes a dictionary (dict) to manage user accounts effectively.

- **Efficiency**: With an average time complexity of O(1) for operations such as lookup, insertion, and deletion, the dictionary allows for quick access to account information. This is crucial for high-frequency banking operations, where every millisecond counts.
- **Scalability**: The dictionary can grow dynamically, accommodating thousands of accounts without significant performance degradation. This scalability is essential as the user base expands.

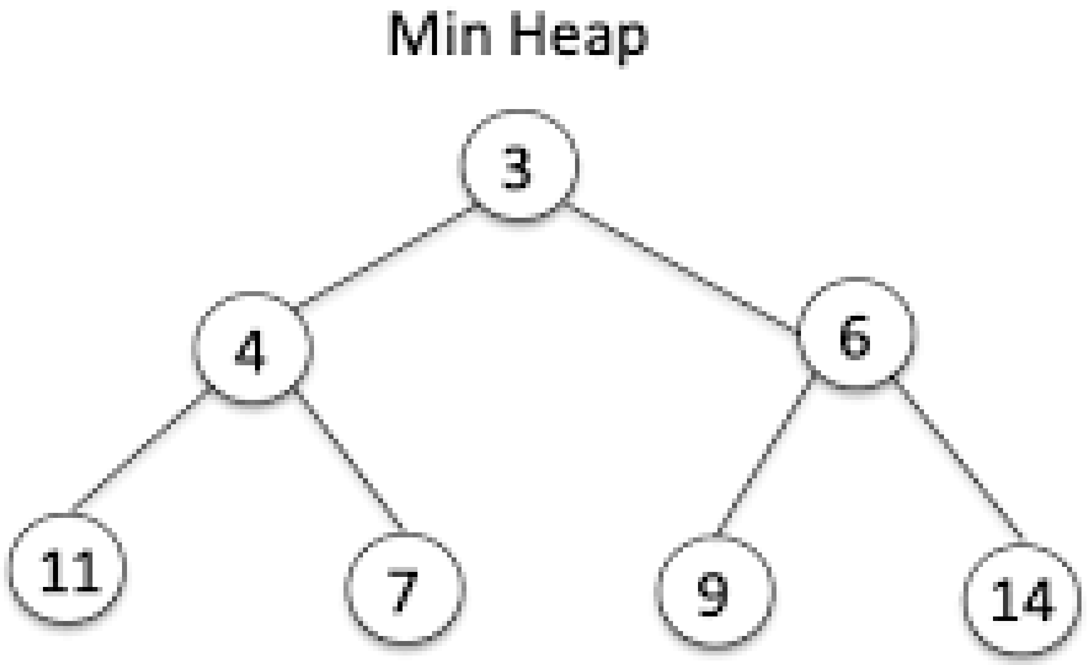
Deque for Transaction History

The use of a double-ended queue (collections.deque) for transaction history management brings several benefits:

- **Constant-Time Operations:** The deque allows for O(1) time complexity for both appending new transactions and popping old ones, making it ideal for maintaining a fixed-size history.
- **Memory Efficiency:** By limiting the size of the deque, the application can effectively manage memory usage, storing only the most recent transactions. This is particularly important for users who perform many transactions.



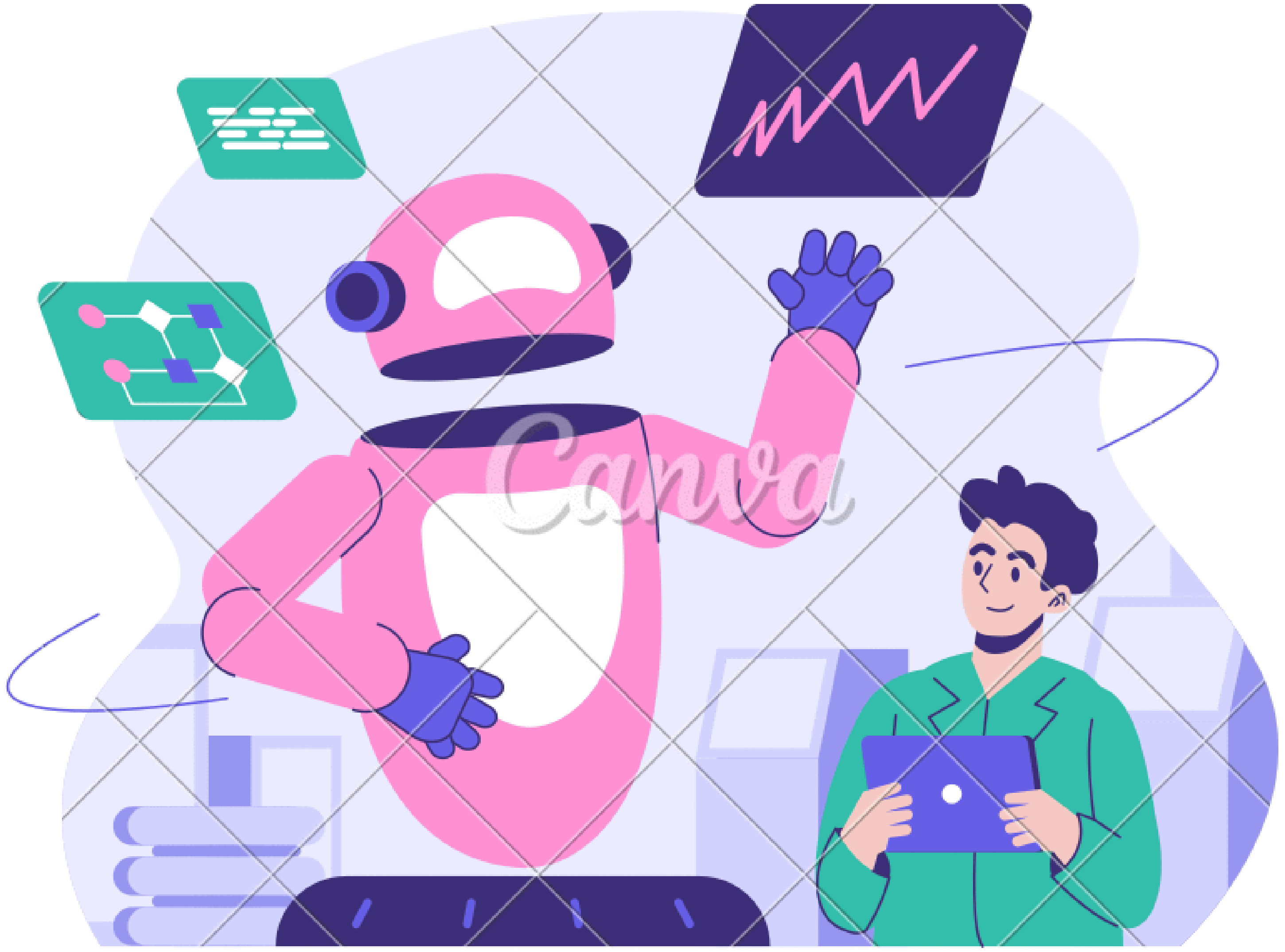
Priority Queue (Heap)



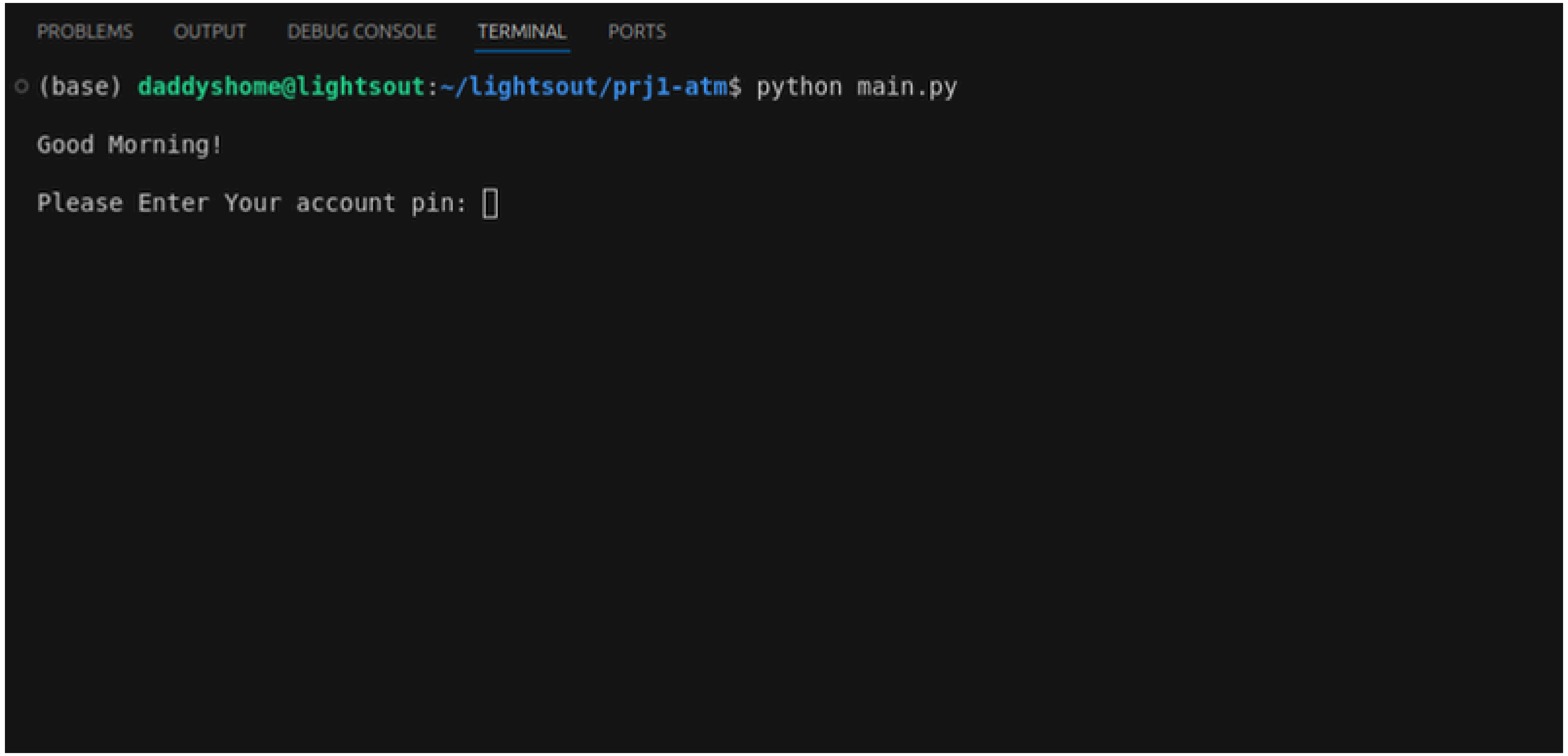
To manage transaction limits and cooldowns, a min-heap (heapq) is implemented.

- **Optimized Scheduling:** The priority queue efficiently tracks the next reset times for each account, ensuring that operations like insertion and extraction occur in O(logN) time.
- **Focused Operations:** By only processing accounts nearing their reset times, the system minimizes unnecessary computations.

IV. RESULTS



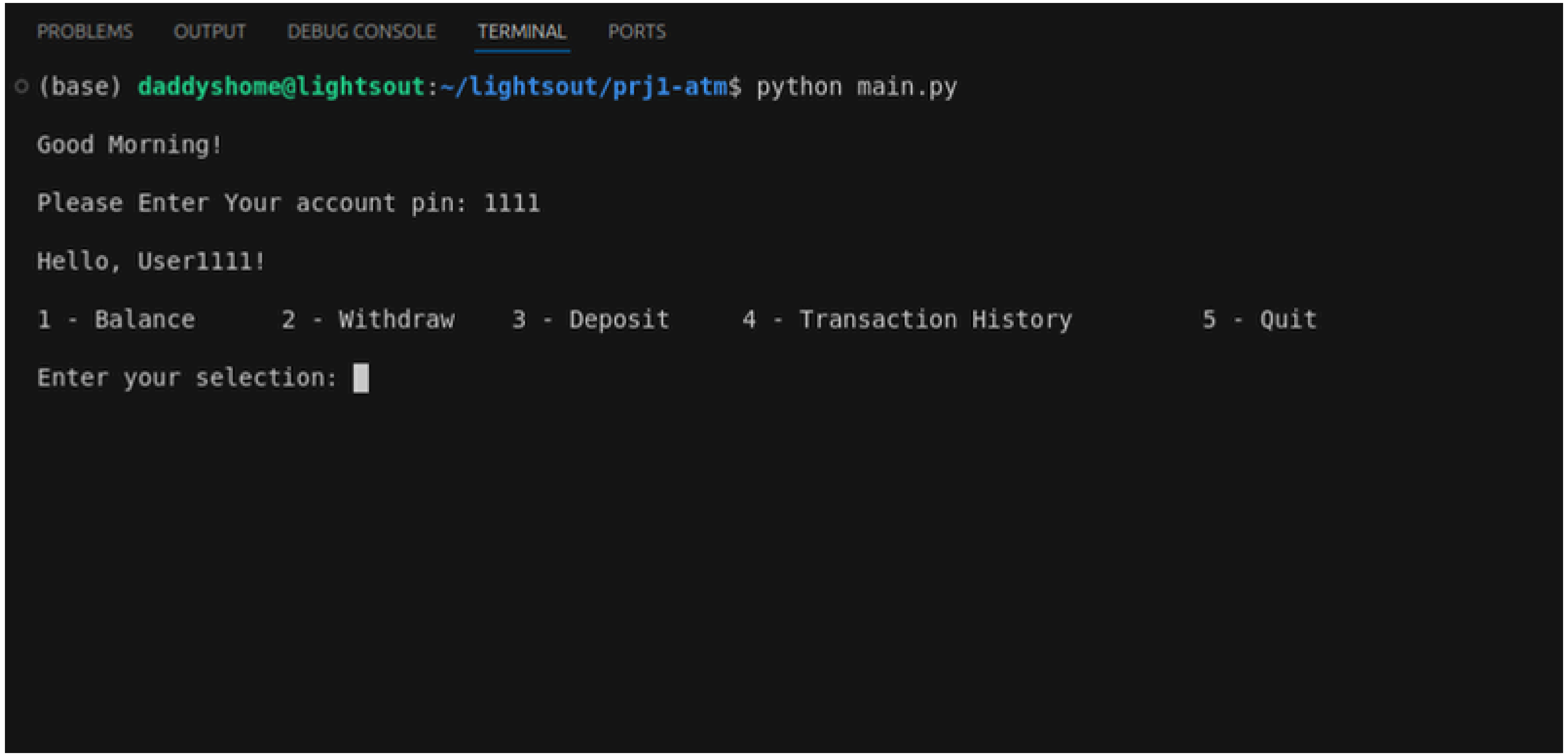
Account Management



```
(base) daddyshome@lightsout:~/lightsout/prj1-atm$ python main.py

Good Morning!

Please Enter Your account pin: 
```

Welcome Menu

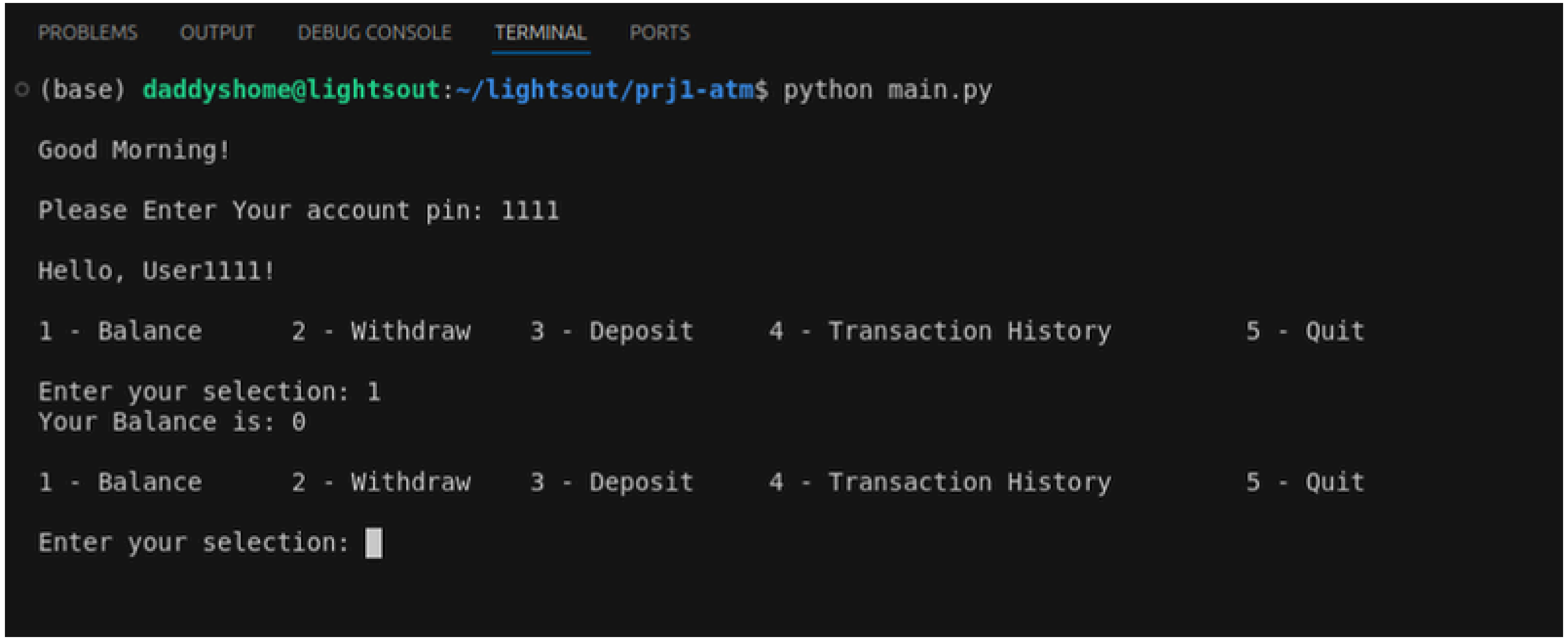
Account Management



```
(base) daddyshome@lightsout:~/lightsout/prj1-atm$ python main.py

Good Morning!

Please Enter Your account pin: 1111

Hello, User1111!

1 - Balance      2 - Withdraw    3 - Deposit     4 - Transaction History         5 - Quit

Enter your selection: 
```

Logging in

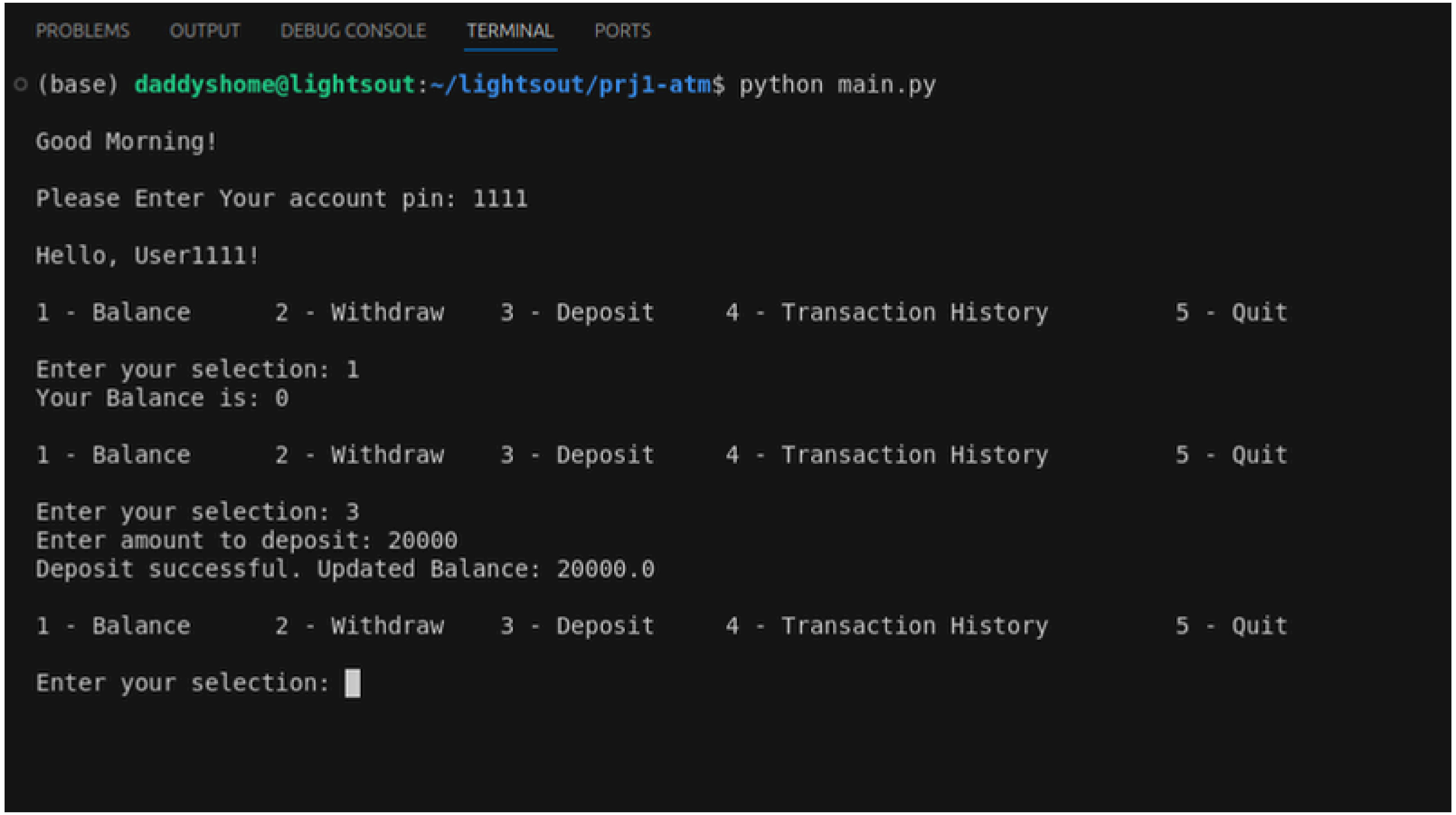
Balance Checking



```
○ (base) daddyshome@lightsout:~/lightsout/prj1-atm$ python main.py

Good Morning!

Please Enter Your account pin: 1111

Hello, User1111!

1 - Balance      2 - Withdraw    3 - Deposit     4 - Transaction History         5 - Quit

Enter your selection: 1
Your Balance is: 0

1 - Balance      2 - Withdraw    3 - Deposit     4 - Transaction History         5 - Quit

Enter your selection: 
```

Checking User Balance

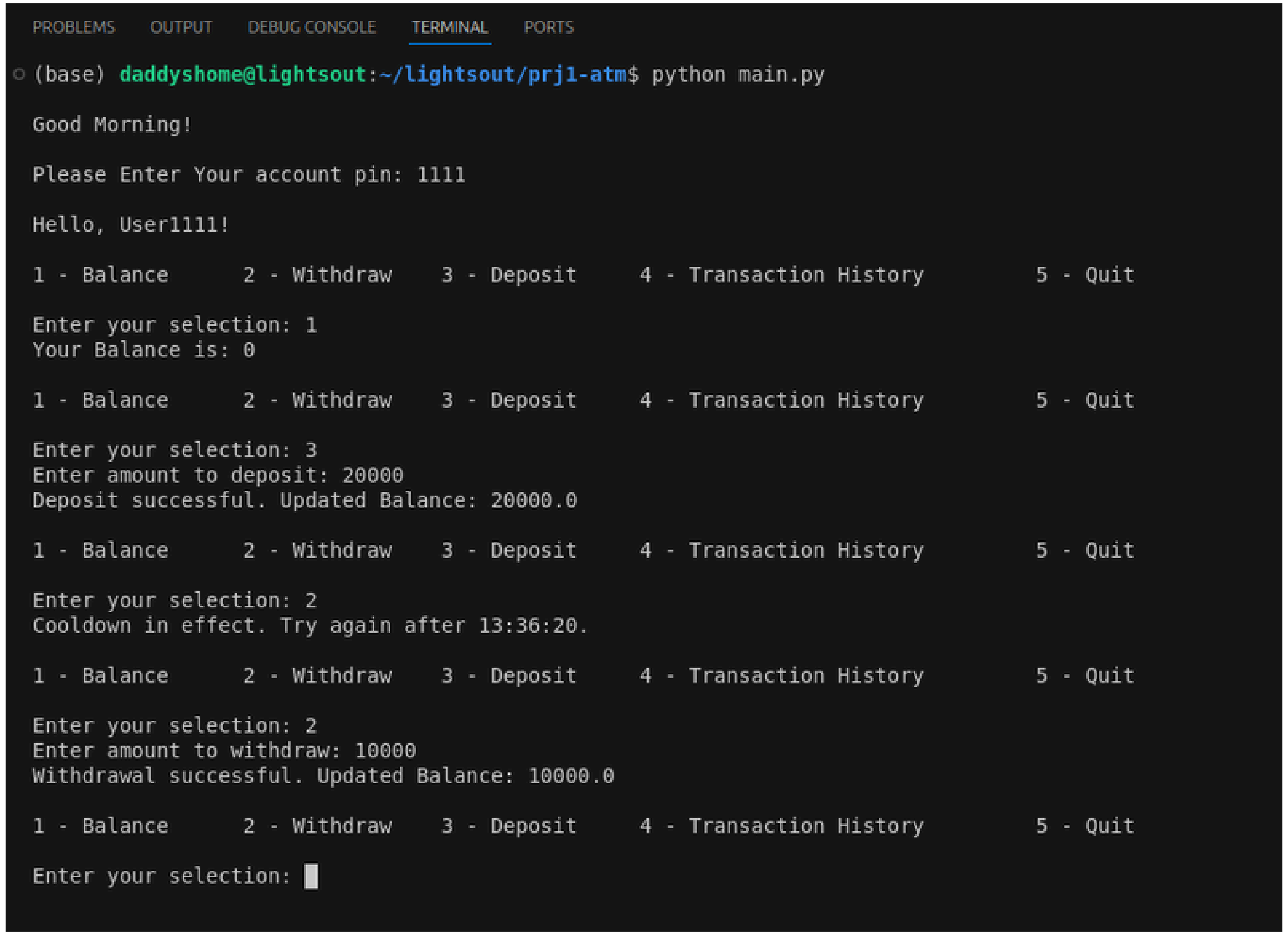
Deposit



```
(base) daddyshome@lightsout:~/lightsout/prj1-atm$ python main.py

Good Morning!

Please Enter Your account pin: 1111

Hello, User1111!

1 - Balance      2 - Withdraw    3 - Deposit     4 - Transaction History         5 - Quit

Enter your selection: 1
Your Balance is: 0

1 - Balance      2 - Withdraw    3 - Deposit     4 - Transaction History         5 - Quit

Enter your selection: 3
Enter amount to deposit: 20000
Deposit successful. Updated Balance: 20000.0

1 - Balance      2 - Withdraw    3 - Deposit     4 - Transaction History         5 - Quit

Enter your selection: 
```

Depositing Money Into Account

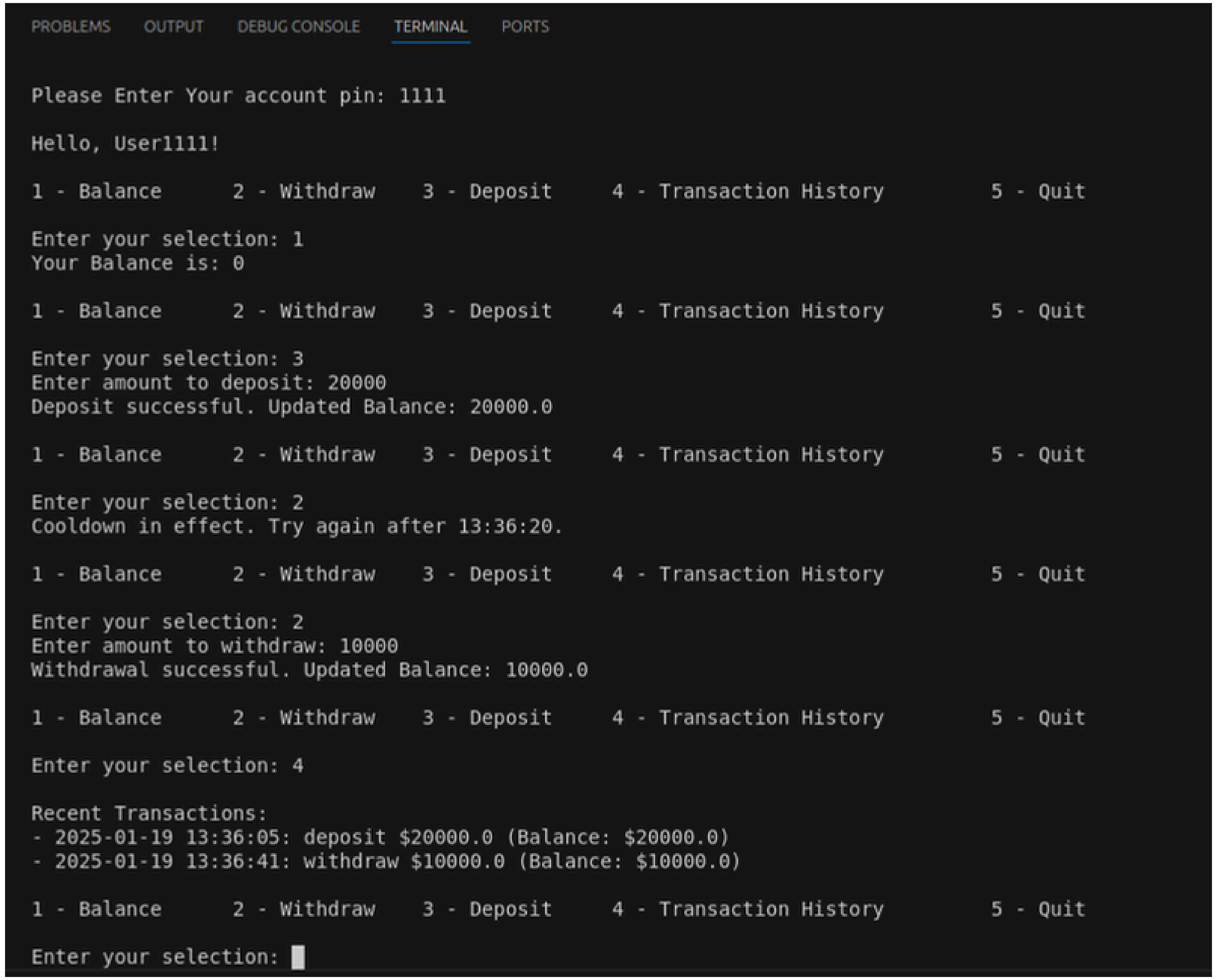
Withdrawal



```
(base) daddyshome@lightsout:~/lightsout/prj1-atm$ python main.py

Good Morning!

Please Enter Your account pin: 1111

Hello, User1111!

1 - Balance      2 - Withdraw    3 - Deposit     4 - Transaction History         5 - Quit

Enter your selection: 1
Your Balance is: 0

1 - Balance      2 - Withdraw    3 - Deposit     4 - Transaction History         5 - Quit

Enter your selection: 3
Enter amount to deposit: 20000
Deposit successful. Updated Balance: 20000.0

1 - Balance      2 - Withdraw    3 - Deposit     4 - Transaction History         5 - Quit

Enter your selection: 2
Cooldown in effect. Try again after 13:36:20.

1 - Balance      2 - Withdraw    3 - Deposit     4 - Transaction History         5 - Quit

Enter your selection: 2
Enter amount to withdraw: 10000
Withdrawal successful. Updated Balance: 10000.0

1 - Balance      2 - Withdraw    3 - Deposit     4 - Transaction History         5 - Quit

Enter your selection: 
```

Withdrawing Money from Account

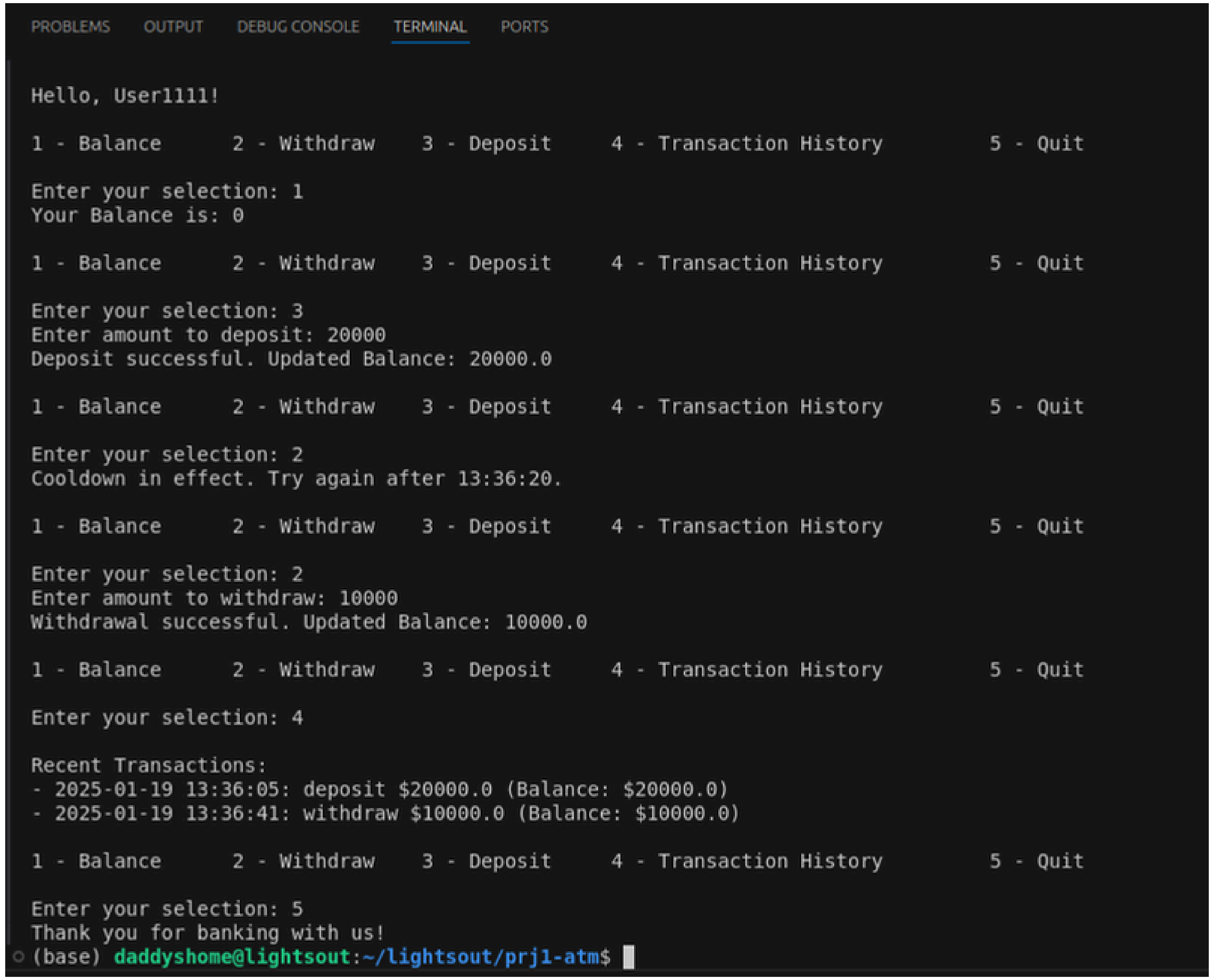
Transaction History



```
Please Enter Your account pin: 1111

Hello, User1111!

1 - Balance      2 - Withdraw    3 - Deposit     4 - Transaction History         5 - Quit

Enter your selection: 1
Your Balance is: 0

1 - Balance      2 - Withdraw    3 - Deposit     4 - Transaction History         5 - Quit

Enter your selection: 3
Enter amount to deposit: 20000
Deposit successful. Updated Balance: 20000.0

1 - Balance      2 - Withdraw    3 - Deposit     4 - Transaction History         5 - Quit

Enter your selection: 2
Cooldown in effect. Try again after 13:36:20.

1 - Balance      2 - Withdraw    3 - Deposit     4 - Transaction History         5 - Quit

Enter your selection: 2
Enter amount to withdraw: 10000
Withdrawal successful. Updated Balance: 10000.0

1 - Balance      2 - Withdraw    3 - Deposit     4 - Transaction History         5 - Quit

Enter your selection: 4

Recent Transactions:
- 2025-01-19 13:36:05: deposit $20000.0 (Balance: $20000.0)
- 2025-01-19 13:36:41: withdraw $10000.0 (Balance: $10000.0)

1 - Balance      2 - Withdraw    3 - Deposit     4 - Transaction History         5 - Quit

Enter your selection: 
```

Viewing Recent Transaction History

Quit and Save Progress



```
Hello, User1111!

1 - Balance      2 - Withdraw    3 - Deposit     4 - Transaction History         5 - Quit

Enter your selection: 1
Your Balance is: 0

1 - Balance      2 - Withdraw    3 - Deposit     4 - Transaction History         5 - Quit

Enter your selection: 3
Enter amount to deposit: 20000
Deposit successful. Updated Balance: 20000.0

1 - Balance      2 - Withdraw    3 - Deposit     4 - Transaction History         5 - Quit

Enter your selection: 2
Cooldown in effect. Try again after 13:36:20.

1 - Balance      2 - Withdraw    3 - Deposit     4 - Transaction History         5 - Quit

Enter your selection: 2
Enter amount to withdraw: 10000
Withdrawal successful. Updated Balance: 10000.0

1 - Balance      2 - Withdraw    3 - Deposit     4 - Transaction History         5 - Quit

Enter your selection: 4

Recent Transactions:
- 2025-01-19 13:36:05: deposit $20000.0 (Balance: $20000.0)
- 2025-01-19 13:36:41: withdraw $10000.0 (Balance: $10000.0)

1 - Balance      2 - Withdraw    3 - Deposit     4 - Transaction History         5 - Quit

Enter your selection: 5
Thank you for banking with us!
(base) daddyshome@lightsout:~/lightsout/prj1-atm$
```

Logging Out Of Account

V. CONCLUSION

Conclusion

- The ATM Simulator is an innovative educational tool that replicates banking operations through a simple console interface.
- Using object-oriented programming, it encapsulates core ATM functionalities, allowing users to perform transactions under realistic constraints.
- Beyond transactions, it fosters understanding of the algorithms and data structures behind modern banking, empowering users to navigate their finances confidently.

THANK YOU!