PROJECT- ATM SYSTEM

(Python Essentials)

NAME- chellarapu.usha sree

—

REG No- 25BHI10128

—

BRANCH- CSE (HI)

--

FACULTY- Sasmita Padhy

- Introduction :

1)This project is designed for understanding of real-world ATM Systems work by applying core Python concepts such as functions, conditional statements, loops , file/module management.

2)Each part of the atm system- login, balance inquiry, deposit and withdrawal is handled by separate modules to make program organized and easy to maintain.

3) The goal of this project is to demonstrate how Python can be used to build simple financial applications that interact with the user, validate inputs, handle errors safely, and update account information dynamically. It also teaches the importance of designing user-friendly interfaces, even in console applications.

- The Problem Statement :

1)Traditional banking requires customers to physically visit a branch for every transaction, which is time-consuming and inefficient. While physical ATMs exist, software simulations are needed to train future developers in secure banking logic, database management, and user interface design.

2) The core problem this project addresses is the need for a reliable and user-friendly system that allows individuals to perform basic banking operations without physically visiting a bank branch. Many users face difficulties managing simple financial tasks due to the absence of accessible digital tools or lack of familiarity with banking technology

.

3) Despite continuous advancements in banking software, people still experience challenges such as long queues at ATMs, errors during transactions, and confusion when interacting with financial systems.

- Functional Requirements:

  1)User login Module : System should verify pin number where user is given a limited number of attempts.

2)Main module: Displays ATM system menu which allows user to choose the task and takes the user to specified service they need from atm.

3)Withdraw module: Allows user to withdraw amount from system and also displays the available balance with a receipt

4)Deposit module: Allows user to deposit money and displays the available balance.

- Non- functional requirements:

1) Usability:

The system should provide a simple and user-friendly console interface.

Menu options must be clearly displayed and easy to navigate for beginners.

Error messages should guide the user with clear instructions (e.g., invalid input, insufficient balance)

2. Performance:

All operations such as login, balance check, deposit, and withdrawal should be processed instantly.

The system should handle input validation quickly without noticeable delays.

3. Reliability :

The system should consistently produce accurate results for all transactions.

It must handle invalid inputs gracefully without crashing.

4. Security:

User authentication must include limited login attempts to prevent unauthorized access.

Sensitive information such as PINs must not be displayed openly on the screen.

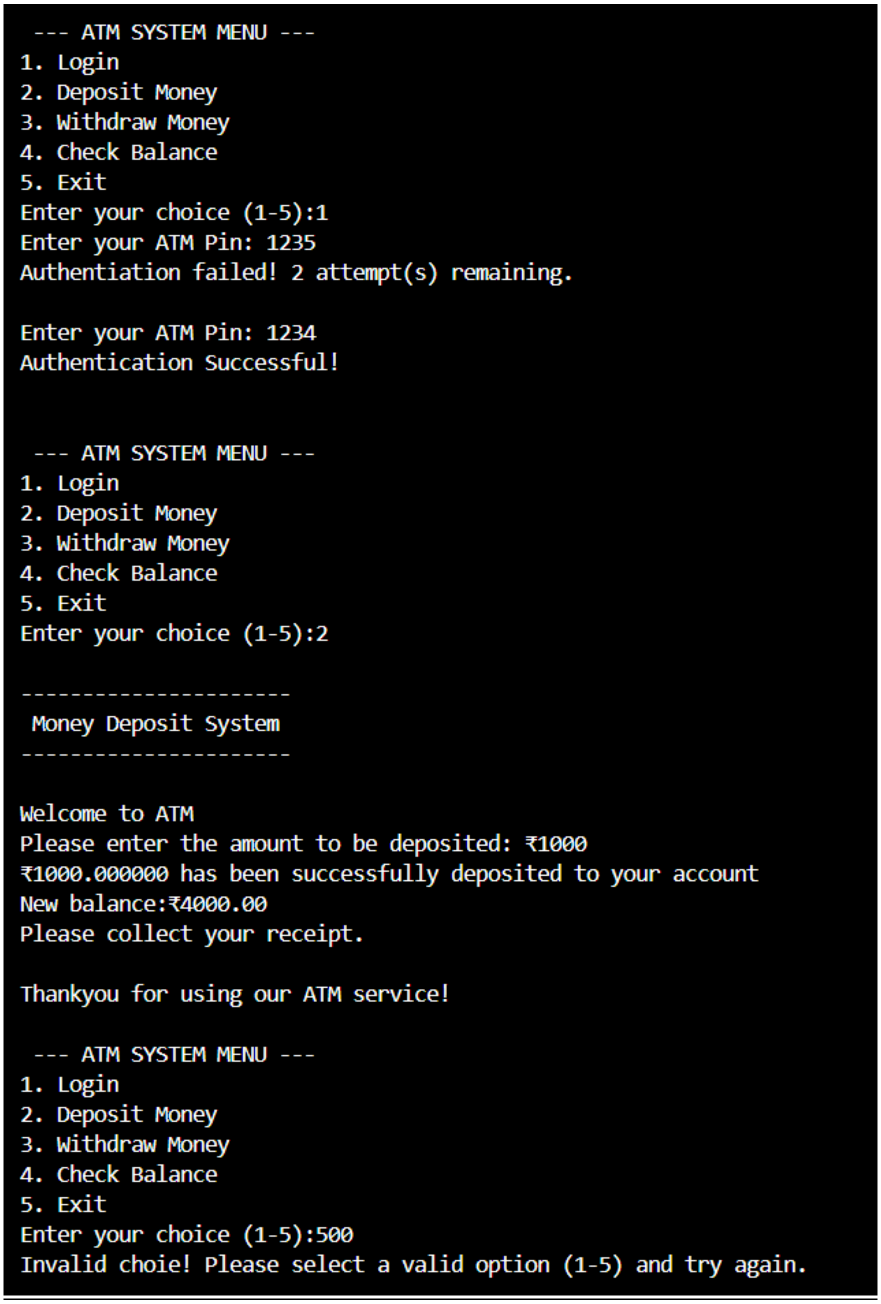
No unauthorized operations should occur without successful login

5. Portability:

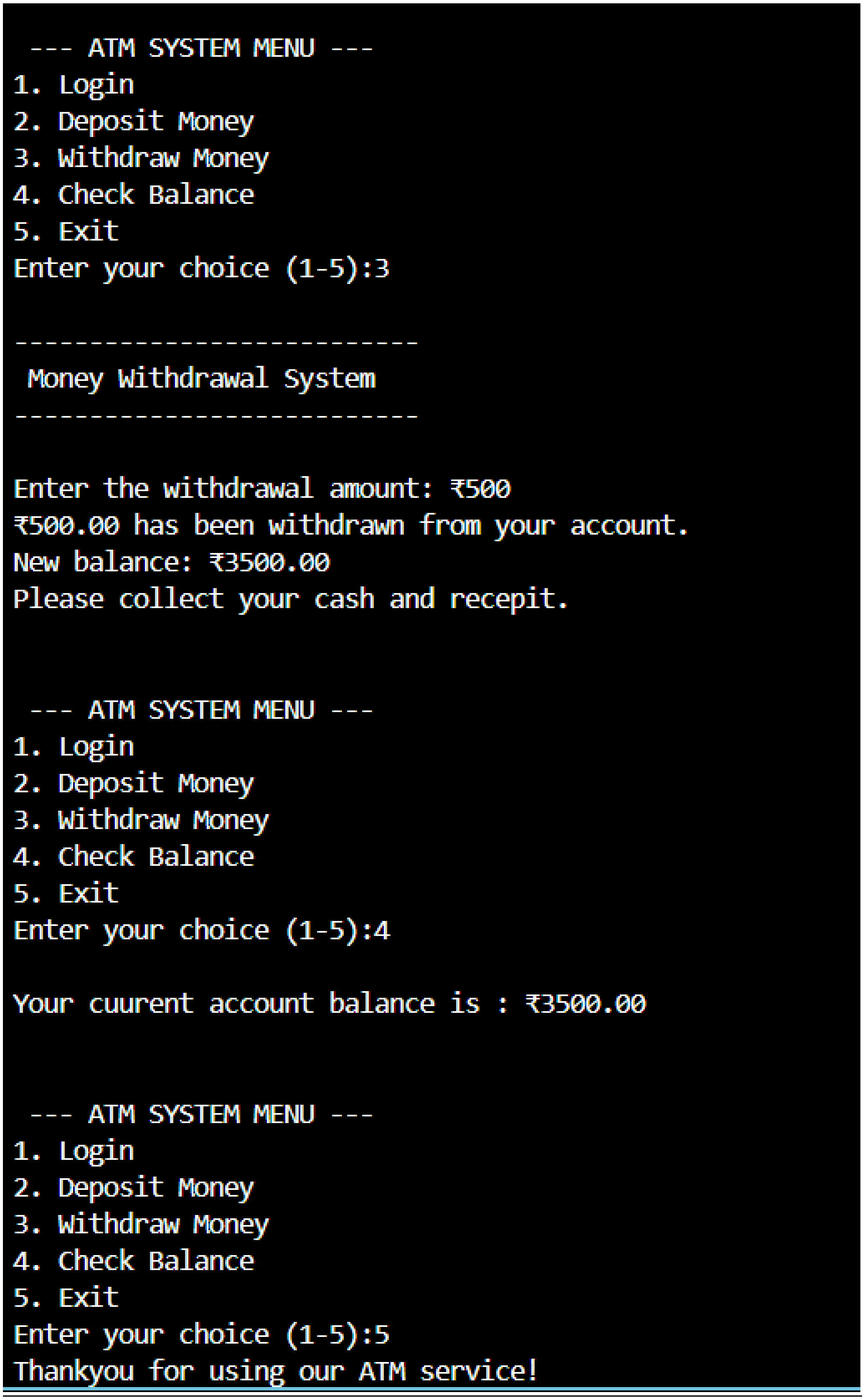
The program should run on all systems supporting Python 3 (Windows, Linux, macOS).

No platform-specific features should be used.

- SCREENSHOT OF OUTPUT:



```
 --- ATM SYSTEM MENU ---
1. Login
2. Deposit Money
3. Withdraw Money
4. Check Balance
5. Exit
Enter your choice (1-5):1
Enter your ATM Pin: 1235
Authentiation failed! 2 attempt(s) remaining.

Enter your ATM Pin: 1234
Authentication Successful!


 --- ATM SYSTEM MENU ---
1. Login
2. Deposit Money
3. Withdraw Money
4. Check Balance
5. Exit
Enter your choice (1-5):2

----------------------
 Money Deposit System
----------------------

Welcome to ATM
Please enter the amount to be deposited: ₹1000
₹1000.000000 has been successfully deposited to your account
New balance:₹4000.00
Please collect your receipt.

Thankyou for using our ATM service!

 --- ATM SYSTEM MENU ---
1. Login
2. Deposit Money
3. Withdraw Money
4. Check Balance
5. Exit
Enter your choice (1-5):500
Invalid choie! Please select a valid option (1-5) and try again.
```



```
 --- ATM SYSTEM MENU ---
1. Login
2. Deposit Money
3. Withdraw Money
4. Check Balance
5. Exit
Enter your choice (1-5):3

---------------------------
 Money Withdrawal System
---------------------------

Enter the withdrawal amount: ₹500
₹500.00 has been withdrawn from your account.
New balance: ₹3500.00
Please collect your cash and recepit.


 --- ATM SYSTEM MENU ---
1. Login
2. Deposit Money
3. Withdraw Money
4. Check Balance
5. Exit
Enter your choice (1-5):4

Your cuurent account balance is : ₹3500.00


 --- ATM SYSTEM MENU ---
1. Login
2. Deposit Money
3. Withdraw Money
4. Check Balance
5. Exit
Enter your choice (1-5):5
Thankyou for using our ATM service!
```

- Learnings & Key Takeaways:

1. Understanding of Core Python Concepts

Developing the ATM System helped reinforce essential Python programming skills, including variables, functions, conditional statements, loops, and exception handling. It provided hands-on experience with writing clean, modular, and reusable code.

2. Improved Problem-Solving Ability

The project enhanced logical thinking by requiring careful planning of how each ATM operation—login, deposit, withdrawal, and balance checking—should behave. It encouraged breaking a complex problem into smaller, manageable components.

3. Modular Programming and Code Organization

By separating features into multiple files or functions, the project demonstrated how modular design improves code readability, maintainability, and scalability—skills essential for larger software applications.

4. Confidence in Building Practical Applications

Completing a functional ATM system boosted confidence in the ability to design and develop real-world applications using Python and to extend them with future features like mini-statements, PIN reset, or multiple accounts.

Future Features :

1. Multi-User Account Support

Allow multiple users to register and maintain their own account balance, PIN, and transaction history. This would make the system more realistic and scalable.

2. PIN Change / Password Reset Option

Add functionality for users to securely change their ATM PIN or reset it through identity verification steps.

3. Mini-Statement / Transaction History

Store and display recent transactions such as deposits, withdrawals, and balance inquiries. This can be implemented using files or a database.

4. Money Transfer Between Accounts

Enable users to transfer funds from their account to another user’s account within the system, simulating real ATM inter-account transfers.

5. Multi-language Support

Add language selection options to improve accessibility for diverse users.