

PROJECT REPORT

INSTRUCTOR:MAA'M TAYYEB A

Submitted by:SAWAIRA MASOOD

BS-CS 1ST semester

Roll no:34

PROJECT TITLE:ATM INTERFACE



INTRODUCTION

- This project is a simulation of an **ATM (Automated Teller Machine) System**, designed to provide users with an interactive and secure banking experience. The program allows users to perform essential banking functions such as creating accounts, checking balances, depositing and withdrawing money, transferring funds, and managing their ATM PIN.
- Unlike a standard ATM interface, this program incorporates **enhanced security features**, including a **two-step verification system** for PIN recovery and automatic account locking after multiple incorrect PIN attempts. It also supports multi-user operations, allowing secure and seamless fund transfers between two accounts.
- With a focus on user-friendly design and robust input validation, the program ensures that all transactions are conducted securely and transparently. Additionally, features like transaction history and low-balance warnings improve the overall usability and practicality of the system.
- This project serves as a comprehensive demonstration of programming fundamentals, such as **structs**, **control flow**, **modularity**, and **file-less data management**, while incorporating practical elements of real-world banking systems. It is an excellent tool for learning about secure coding practices and enhancing problem-solving skills in programming.

➤ TECHNOLOGY STACK

- **Programming Language:** C++
- **Development Environment:** Any C++ IDE or compiler (e.g., Code::Blocks, Visual Studio, GCC).

➤ WORKFLOW

1. **Initialization:**
 - Users are prompted to create accounts before performing any other operations.
2. **Authentication:**
 - PIN-based authentication ensures that only authorized users can access account features.
3. **Transaction Execution:**
 - Based on user input, the appropriate function is executed, with real-time validation and updates to account details.

FEATURES

1. **Create Account:**

- Users can create accounts for themselves and another person by entering details like name, PIN, two-step verification code, and initial balance.

2. **Deposit Money:**

- Users can securely deposit money into their account by entering their PIN.

3. **Withdraw Money:**

- Users can withdraw money securely by entering their PIN and specifying the amount.

4. **Check Balance:**

- Users can view their current account balance. A warning is issued for low balances.

Here are the **additional features** i have added to program beyond the basic ATM functionalities:

1. TWO-STEP VERIFICATION FOR PIN RECOVERY

- A **two-step verification mechanism** allows users to reset their ATM PIN securely.
- During account creation, users set a **4-digit verification code**, which is required for PIN recovery.
- This feature enhances the security of the program, providing a safety net for forgotten PINs.

2. ACCOUNT LOCKING FOR SECURITY

- After **three consecutive incorrect PIN attempts**, the account is automatically locked.
- This prevents unauthorized access and enhances the security of the account.

3. LOW BALANCE WARNING

- When users check their balance, the program provides a **low-balance warning** if the balance falls below a certain threshold (e.g., less than 100 units).
- This feature ensures that users are notified to maintain sufficient funds.

4. ENHANCED TRANSACTION HISTORY

- The transaction history now includes detailed records for:
 - Deposits.
 - Withdrawals.
 - Fund transfers (with the recipient or sender's name mentioned).
- This provides users with a clearer understanding of their account activity.

- Extensive **input validation** ensures:
 - Deposits are greater than zero.
 - Withdrawals are within the available balance.
 - Transfer amounts are valid and within the sender's account balance.

6. MULTI-USER SUPPORT

- The program supports the creation and management of **two accounts**:
 - A primary user account.
 - A secondary user account (for fund transfers).
- This adds a multi-user dynamic for fund transfer operations.

7. SECURE FUND TRANSFER

- Added functionality to transfer funds between two accounts with:
 - Validation for sufficient balance in the sender's account.
 - Real-time updates to both accounts' transaction history and balances.

8 FAILED ATTEMPT WARNINGS

- During PIN validation for deposits, withdrawals, and transfers:
 - Users receive a warning showing the number of remaining attempts.
 - If all attempts are exhausted, a **lockout message** is displayed.

OUTPUT

```

=====ATM Menu=====
1.Create an Account for yourself
2. Create Account for aother person
3. Deposit Money
4. Withdraw Money
5. Check your Balance
6. View your Transaction History
7. Change your ATM Pin
8. Recover your ATM PIN
9. Transfer Funds to another account
10. Exit

Which action do you want to perform: 1
Enter name to create account: sawera
Enter a 4-digit PIN for account: 1234
Enter a 4-digit two step verification code for password recovery: 0000
Enter initial balance: 256379
=====ATM Menu=====
1.Create an Account for yourself
2. Create Account for aother person
3. Deposit Money
4. Withdraw Money
5. Check your Balance
6. View your Transaction History
7. Change your ATM Pin
8. Recover your ATM PIN
9. Transfer Funds to another account
10. Exit

Which action do you want to perform: 2
Enter name to create account: honey
Enter a 4-digit PIN for account: 3333
Enter a 4-digit two step verification code for password recovery: 8765
Enter initial balance: 253781

```

```

=====ATM Menu=====
1.Create an Account for yourself
2. Create Account for aother person
3. Deposit Money
4. Withdraw Money
5. Check your Balance
6. View your Transaction History
7. Change your ATM Pin
8. Recover your ATM PIN
9. Transfer Funds to another account
10. Exit

Which action do you want to perform: 3
Enter your PIN to deposit money: 1234
Enter the amount you want to deposit: 89000
Deposit successful!
After this deposit, your new balance is: 345379
=====ATM Menu=====
1.Create an Account for yourself
2. Create Account for aother person
3. Deposit Money
4. Withdraw Money
5. Check your Balance
6. View your Transaction History
7. Change your ATM Pin
8. Recover your ATM PIN
9. Transfer Funds to another account
10. Exit

Which action do you want to perform: 4
Enter your PIN to withdraw money: 52672
Incorrect PIN! You have 2 attempt(s) left.
Enter your PIN to withdraw money: 1723
Incorrect PIN! You have 1 attempt(s) left.
Enter your PIN to withdraw money: 1234
Enter the amount that you want to withdraw from your account: 4367
Withdrawal successful!
Your new balance after this withdrawal is: 341012

```

- In the **first picture**, the operations **Create an Account for yourself** and **Create Account for another person** are performed, and their actions are displayed.
- In the **second picture**, the operations **Deposit Money** and **Withdraw Money** are executed, with their corresponding actions shown.

```

#####
-----ATM Menu-----
1.Create an Account for yourself
2. Create Account for aother person
3. Deposit Money
4. Withdraw Money
5. Check your Balance
6. View your Transaction History
7. Change your ATM Pin
8. Recover your ATM PIN
9. Transfer Funds to another account
10. Exit

#####

Which action do you want to perform: 5
Current balance: 341012
#####
-----ATM Menu-----
1.Create an Account for yourself
2. Create Account for aother person
3. Deposit Money
4. Withdraw Money
5. Check your Balance
6. View your Transaction History
7. Change your ATM Pin
8. Recover your ATM PIN
9. Transfer Funds to another account
10. Exit

#####

Which action do you want to perform: 6
Transaction History:
Account created with balance: 256379.000000Deposited: 89000.000000Withdrawn: 4367.000000Total Transactions: 2
#####

```

```

2. Create Account for aother person
3. Deposit Money
4. Withdraw Money
5. Check your Balance
6. View your Transaction History
7. Change your ATM Pin
8. Recover your ATM PIN
9. Transfer Funds to another account
10. Exit

#####

Which action do you want to perform: 7
Enter current PIN: 1234
Enter new PIN: 5678
PIN changed successfully!
#####
-----ATM Menu-----
1.Create an Account for yourself
2. Create Account for aother person
3. Deposit Money
4. Withdraw Money
5. Check your Balance
6. View your Transaction History
7. Change your ATM Pin
8. Recover your ATM PIN
9. Transfer Funds to another account
10. Exit

#####

Which action do you want to perform: 8
Enter Two step verification code to reset PIN: 0000
Enter new PIN: 1234
PIN reset successfully!Your new PIN now to onward is:1234
#####

```

```

#####
-----ATM Menu-----
1.Create an Account for yourself
2. Create Account for aother person
3. Deposit Money
4. Withdraw Money
5. Check your Balance
6. View your Transaction History
7. Change your ATM Pin
8. Recover your ATM PIN
9. Transfer Funds to another account
10. Exit

#####

Which action do you want to perform: 9
Enter amount to transfer: 4567
Transfer successful! after this transaction your new balance is: 336445
#####
-----ATM Menu-----
1.Create an Account for yourself
2. Create Account for aother person
3. Deposit Money
4. Withdraw Money
5. Check your Balance
6. View your Transaction History
7. Change your ATM Pin
8. Recover your ATM PIN
9. Transfer Funds to another account
10. Exit

#####

Which action do you want to perform: 10
Exiting... Thank you for using the ATM!
-----
Process exited after 370.2 seconds with return value 0
Press any key to continue . . .

```

- In the **first picture**, the operations Deposit Money and Withdraw Money and their actions are displayed.
- In the **second picture**, the operations Check your Balance and View your Transaction History are executed, with their corresponding actions shown.
- In the **third picture**, the last two operations are executed, with their corresponding actions shown.

SOURCE CODE

```
#include <iostream>
#include <string>
using namespace std;

struct Account {
    string name;
    int pin;
    double balance;
    string history;
    int countTransaction;
    int NumOfAttempts;
    int TwoStepVerification;
};

void createAccount(Account &acc) {
    cout << "Enter name to create account: ";
    cin >> acc.name;
    cout << "Enter a 4-digit PIN for account: ";
    cin >> acc.pin;
    cout << "Enter a 4-digit two step verification code for password recovery: ";
    cin >> acc.TwoStepVerification;
    cout << "Enter initial balance: ";
    cin >> acc.balance;
    acc.history = "Account created with balance: " + to_string(acc.balance);
    acc.countTransaction = 0;
    acc.NumOfAttempts = 0;
}

void deposit(Account &acc) {
    double amount;
    int pin;
    int maxAttempts = 3;
    int attempts = 0;
    while (attempts < maxAttempts) {
        cout << "Enter your PIN to deposit money: ";
        cin >> pin;
        if (pin == acc.pin) {
            cout << "Enter the amount you want to deposit: ";
            cin >> amount;
            if (amount > 0) {
                acc.balance += amount;
                acc.history += "Deposited: " + to_string(amount);
                acc.countTransaction++;
                cout << "\nDeposit successful!";
            }
        }
        else {
            attempts++;
            cout << "Incorrect PIN! You have " << (maxAttempts - attempts)
                << " attempt(s) left." << endl;
        }
    }
    if (attempts == maxAttempts) {
        cout << "Maximum attempts reached! Withdrawal canceled for security reasons & account is locked\n";
    }
}

void checkBalance(const Account &acc) {
    cout << "Current balance: " << acc.balance << endl;
    if (acc.balance < 100) {
        cout << "Warning: Low balance! Please upgrade your balance" << endl;
    }
}

void history(const Account &acc) {
    cout << "Transaction History:\n" << acc.history;
    cout << "Total Transactions: " << acc.countTransaction << endl;
}

void changePin(Account &acc) {
    int oldPin, newPin;
    cout << "Enter current PIN: ";
    cin >> oldPin;
    if (oldPin == acc.pin) {
        cout << "Enter new PIN: ";
        cin >> newPin;
        acc.pin = newPin;
        cout << "PIN changed successfully!" << endl;
    }
    else {
        acc.NumOfAttempts++;
        cout << "You enter the wrong PIN, Please enter the correct one!" << endl;
        if (acc.NumOfAttempts == 3) {
            cout << "Warning: Account is locked due to multiple incorrect attempts!" << endl;
        }
    }
}

void recoverPin(Account &acc) {
    int code;
    cout << "Enter Two step verification code to reset PIN: ";
    cin >> code;
    if (code == acc.TwoStepVerification) {
        int newPin;
        cout << "Enter new PIN: ";
        cin >> newPin;
        acc.pin = newPin;
        cout << "PIN reset successfully!" << "Your new PIN now onward is: " << newPin << endl;
    }
}
```

```
100 void changePin(Account &acc) {
101     int oldPin, newPin;
102     cout << "Enter current PIN: ";
103     cin >> oldPin;
104     if (oldPin == acc.pin) {
105         cout << "Enter new PIN: ";
106         cin >> newPin;
107         acc.pin = newPin;
108         cout << "PIN changed successfully!" << endl;
109     }
110     else {
111         acc.NumOfAttempts++;
112         cout << "You enter the wrong PIN, Please enter the correct one!" << endl;
113         if (acc.NumOfAttempts == 3) {
114             cout << "Warning: Account is locked due to multiple incorrect attempts!" << endl;
115         }
116     }
117 }

118 void recoverPin(Account &acc) {
119     int code;
120     cout << "Enter Two step verification code to reset PIN: ";
121     cin >> code;
122     if (code == acc.TwoStepVerification) {
123         int newPin;
124         cout << "Enter new PIN: ";
125         cin >> newPin;
126         acc.pin = newPin;
127         cout << "PIN reset successfully!" << "Your new PIN now onward is: " << newPin << endl;
128     }
129 }
```

```
43     cout << "\nAfter this deposit, your new balance is: " << acc.balance << endl;
44     return;
45 }
46 } else {
47     cout << "Invalid amount! Deposit must be greater than 0." << endl;
48     return;
49 }
50 } else {
51     attempts++;
52     cout << "Incorrect PIN! You have " << (maxAttempts - attempts)
53         << " attempt(s) left." << endl;
54 }
55 }
56 cout << "Maximum attempts reached! Deposit canceled for security reasons & account is locked!!!." << endl;
57 }

58 void withdraw(Account &acc) {
59     double amount;
60     int pin;
61     int maxAttempts = 3;
62     int attempts = 0;
63     while (attempts < maxAttempts) {
64         cout << "Enter your PIN to withdraw money: ";
65         cin >> pin;
66         if (pin == acc.pin) {
67             cout << "Enter the amount that you want to withdraw from your account: ";
68             cin >> amount;
69             if (amount > 0 && amount <= acc.balance) {
70                 acc.balance -= amount;
71                 acc.history += "Withdrawn: " + to_string(amount);
72                 acc.countTransaction++;
73                 cout << "\nWithdrawal successful!";
74                 cout << "\nYour new balance after this withdrawal is: " << acc.balance << endl;
75                 return;
76             }
77             else if (amount > acc.balance) {
78                 cout << "Invalid amount! Your balance is insufficient." << endl;
79                 return;
80             }
81             else {
82                 cout << "Invalid amount! Withdrawal amount must be greater than 0." << endl;
83                 return;
84             }
85         }
86         else {
87             attempts++;
88             cout << "Incorrect PIN! You have " << (maxAttempts - attempts)
89                 << " attempt(s) left." << endl;
90         }
91     }
92     if (attempts == maxAttempts) {
93         cout << "Maximum attempts reached! Withdrawal canceled for security reasons & account is locked\n";
94     }
95 }
```

```
130 }

131 void transferFunds(Account &sender, Account &receiver) {
132     double amount;
133     cout << "Enter amount to transfer: ";
134     cin >> amount;
135     if (amount > 0 && amount <= sender.balance) {
136         sender.balance -= amount;
137         receiver.balance += amount;
138         sender.history += "Transferred: " + to_string(amount) + " to " + receiver.name;
139         receiver.history += "Received: " + to_string(amount) + " from " + sender.name;
140         sender.countTransaction++;
141         receiver.countTransaction++;
142         cout << "Transfer successful! After this transaction your new balance is: " << sender.balance << endl;
143     }
144     else if (amount > sender.balance) {
145         cout << "Insufficient balance!" << endl;
146     }
147     else {
148         cout << "Invalid amount!" << endl;
149     }
150 }

151 int main() {
152     Account acc1, acc2;
153     int choice;
154     bool acc1Created = false, acc2Created = false;
155     do {
156         cout << "=====ATM Menu=====\n";
157         cout << "1. Create Account for yourself ";
158         cout << "2. Create Account for another person ";
159         cout << "3. Deposit Money ";
160         cout << "4. Withdraw Money ";
161         cout << "5. Check your Balance ";
162         cout << "6. View your Transaction History ";
163         cout << "7. Change your ATM Pin ";
164         cout << "8. Recover your ATM PIN ";
165         cout << "9. Transfer Funds to another account ";
166         cout << "10. Exit ";
167         cout << "\n";
168         cout << "=====\n";
169         cout << "\n Which action do you want to perform: ";
170         cin >> choice;
171         switch (choice) {
172             case 1:
173                 if (!acc1Created) {
174                     createAccount(acc1);
175                     acc1Created = true;
176                 }
177                 else {
178                     cout << "You already created an account for yourself!!!" << endl;
179                 }
180                 break;
181             case 2:
182                 if (!acc2Created) {
183                     createAccount(acc2);
184                     acc2Created = true;
185                 }
186                 else {
187                     cout << "For another user account is already created!" << endl;
188                 }
189                 break;
190             case 3:
191                 if (acc1Created) {
192                     deposit(acc1);
193                 }
194                 else {
195                     cout << "No such account found! Please create it first!" << endl;
196                 }
197                 break;
198             case 4:
199                 if (acc1Created) {
200                     withdraw(acc1);
201                 }
202                 else {
203                     cout << "No matching account found! Please create it first!" << endl;
204                 }
205                 break;
206             case 5:
207                 if (acc1Created) {
208                     checkBalance(acc1);
209                 }
210                 else {
211                     cout << "Account not found! Please create it first!" << endl;
212                 }
213                 break;
214             case 6:
215                 history(acc1);
216             case 7:
217                 changePin(acc1);
218             case 8:
219                 recoverPin(acc1);
220             case 9:
221                 transferFunds(acc1, acc2);
222             case 10:
223                 return 0;
224             default:
225                 cout << "Invalid choice! Please enter a valid choice." << endl;
226         }
227     } while (choice != 10);
228 }
```

```
172 switch (choice) {
173     case 1:
174         if (!acc1Created) {
175             createAccount(acc1);
176             acc1Created = true;
177         }
178         else {
179             cout << "You already created an account for yourself!!!" << endl;
180         }
181         break;
182     case 2:
183         if (!acc2Created) {
184             createAccount(acc2);
185             acc2Created = true;
186         }
187         else {
188             cout << "For another user account is already created!" << endl;
189         }
190         break;
191     case 3:
192         if (acc1Created) {
193             deposit(acc1);
194         }
195         else {
196             cout << "No such account found! Please create it first!" << endl;
197         }
198         break;
199     case 4:
200         if (acc1Created) {
201             withdraw(acc1);
202         }
203         else {
204             cout << "No matching account found! Please create it first!" << endl;
205         }
206         break;
207     case 5:
208         if (acc1Created) {
209             checkBalance(acc1);
210         }
211         else {
212             cout << "Account not found! Please create it first!" << endl;
213         }
214         break;
215     case 6:
216         history(acc1);
217     case 7:
218         changePin(acc1);
219     case 8:
220         recoverPin(acc1);
221     case 9:
222         transferFunds(acc1, acc2);
223     case 10:
224         return 0;
225     default:
226         cout << "Invalid choice! Please enter a valid choice." << endl;
227 }
```