Python Bank Account Management System

```
import pickle
import os
import random
import numpy as np
from datetime import datetime

# File to store account data
DATA_FILE = 'accounts_data.pkl'
TRANSACTIONS_FILE = 'transactions_data.pkl'

# Initialize data
accounts = {}
transactions = []

# Load data from files
if os.path.exists(DATA_FILE):
    with open(DATA_FILE, 'rb') as file:
        accounts = pickle.load(file)

if os.path.exists(TRANSACTIONS_FILE):
    with open(TRANSACTIONS_FILE, 'rb') as file:
        transactions = pickle.load(file)

def generate_account_number():
    return str(random.randint(10**10, 10**11 - 1))

def create_password():
    return str(random.randint(100000, 999999))

def save_data():
    with open(DATA_FILE, 'wb') as file:
        pickle.dump(accounts, file)
    with open(TRANSACTIONS_FILE, 'wb') as file:
        pickle.dump(transactions, file)

def record_transaction(account_no, type_, amount,
target_account=None):
    transaction = {
        'account_no': account_no,
        'type': type_,
        'amount': amount,
        'date': datetime.now().strftime('%Y-%m-%d %H:%M:%S'),
        'target_account': target_account
    }
    transactions.append(transaction)
    save_data()
```

```python
# Main Functions
def open_account():
    name = input("Enter account holder's name: ")
    account_type = input("Enter account type (Savings/Current):
").capitalize()
    initial_balance = float(input("Enter initial balance: "))

    account_no = generate_account_number()
    password = create_password()
    user_id = name.lower().replace(' ', '_') + account_no[-4:]

    accounts[account_no] = {
        'name': name,
        'type': account_type,
        'balance': initial_balance,
        'password': password,
        'user_id': user_id
    }
    save_data()
    print(f"Account created successfully! Account No: {account_no},
User ID: {user_id}, Password: {password}")

def view_account():
    account_no = input("Enter account number: ")
    password = input("Enter account password: ")
    account = accounts.get(account_no)

    if account and account['password'] == password:
        print(f"Account Holder: {account['name']}")
        print(f"Account Type: {account['type']}")
        print(f"Account Balance: {account['balance']}")
    else:
        print("Invalid account number or password.")

def deposit():
    account_no = input("Enter account number: ")
    amount = float(input("Enter amount to deposit: "))

    if account_no in accounts and amount > 0:
        accounts[account_no]['balance'] += amount
        record_transaction(account_no, 'Deposit', amount)
        print(f"Deposited {amount} successfully! New Balance:
{accounts[account_no]['balance']}")
    else:
        print("Invalid account number or amount.")

def withdraw():
    account_no = input("Enter account number: ")
    amount = float(input("Enter amount to withdraw: "))
```

```
    if account_no in accounts and 0 < amount <= accounts[account_no]
['balance']:
        accounts[account_no]['balance'] -= amount
        record_transaction(account_no, 'Withdrawal', amount)
        print(f"Withdrew {amount} successfully! New Balance:
{accounts[account_no]['balance']}")
    else:
        print("Invalid account number or insufficient funds.")

def transfer():
    sender_account = input("Enter sender account number: ")
    recipient_account = input("Enter recipient account number: ")
    amount = float(input("Enter amount to transfer: "))

    if sender_account in accounts and recipient_account in accounts
and accounts[sender_account]['balance'] >= amount and amount > 0:
        accounts[sender_account]['balance'] -= amount
        accounts[recipient_account]['balance'] += amount
        record_transaction(sender_account, 'Transfer Out', amount,
recipient_account)
        record_transaction(recipient_account, 'Transfer In', amount,
sender_account)
        print(f"Transferred {amount} from {sender_account} to
{recipient_account} successfully!")
    else:
        print("Invalid account numbers or insufficient funds.")

def view_transaction_history():
    account_no = input("Enter account number: ")
    history = [t for t in transactions if t['account_no'] ==
account_no]

    if history:
        for t in history:
            print(f"Date: {t['date']}, Type: {t['type']}, Amount:
{t['amount']}, Target Account: {t.get('target_account', 'N/A')}")
    else:
        print("No transactions found for this account.")

def summary_statistics():
    account_no = input("Enter account number: ")
    history = [t for t in transactions if t['account_no'] ==
account_no]

    if history:
        deposits = [t['amount'] for t in history if t['type'] ==
'Deposit']
        withdrawals = [t['amount'] for t in history if t['type'] ==
'Withdrawal']
```

```
        print(f"Total Deposits: {sum(deposits)}")
        print(f"Total Withdrawals: {sum(withdrawals)}")
        print(f"Average Transaction Amount: {np.mean([t['amount'] for
t in history])}")
    else:
        print("No transactions found for this account.")

# Main Menu
def main_menu():
    while True:
        print("\nBank Account Management System")
        print("1. Open a New Account")
        print("2. View Account Details")
        print("3. Deposit Money")
        print("4. Withdraw Money")
        print("5. Transfer Money")
        print("6. View Transaction History")
        print("7. View Summary Statistics")
        print("8. Exit")

        choice = input("Enter your choice: ")
        if choice == '1':
            open_account()
        elif choice == '2':
            view_account()
        elif choice == '3':
            deposit()
        elif choice == '4':
            withdraw()
        elif choice == '5':
            transfer()
        elif choice == '6':
            view_transaction_history()
        elif choice == '7':
            summary_statistics()
        elif choice == '8':
            print("Exiting program. Goodbye!")
            break
        else:
            print("Invalid choice. Please try again.")

# Start the program
main_menu()


Bank Account Management System
1. Open a New Account
2. View Account Details
3. Deposit Money
4. Withdraw Money
```

```
5. Transfer Money
6. View Transaction History
7. View Summary Statistics
8. Exit

Enter your choice:  1
Enter account holder's name:  Anchal Dayal
Enter account type (Savings/Current):  Savings
Enter initial balance:  20000

Account created successfully! Account No: 54778603772, User ID:
anchal_dayal3772, Password: 967269

Bank Account Management System
1. Open a New Account
2. View Account Details
3. Deposit Money
4. Withdraw Money
5. Transfer Money
6. View Transaction History
7. View Summary Statistics
8. Exit

Enter your choice:  1
Enter account holder's name:  Khushi Agarwal
Enter account type (Savings/Current):  Savings
Enter initial balance:  800000

Account created successfully! Account No: 31058802008, User ID:
khushi_agarwal2008, Password: 511702

Bank Account Management System
1. Open a New Account
2. View Account Details
3. Deposit Money
4. Withdraw Money
5. Transfer Money
6. View Transaction History
7. View Summary Statistics
8. Exit

Enter your choice:  2
Enter account number:  31058802008
Enter account password:  511702

Account Holder: Khushi Agarwal
Account Type: Savings
Account Balance: 800000.0

Bank Account Management System
1. Open a New Account
```

```
2. View Account Details
3. Deposit Money
4. Withdraw Money
5. Transfer Money
6. View Transaction History
7. View Summary Statistics
8. Exit

Enter your choice:  3
Enter account number:  511702
Enter amount to deposit:  8900

Invalid account number or amount.

Bank Account Management System
1. Open a New Account
2. View Account Details
3. Deposit Money
4. Withdraw Money
5. Transfer Money
6. View Transaction History
7. View Summary Statistics
8. Exit

Enter your choice:  3
Enter account number:  31058802008
Enter amount to deposit:  2000

Deposited 2000.0 successfully! New Balance: 802000.0

Bank Account Management System
1. Open a New Account
2. View Account Details
3. Deposit Money
4. Withdraw Money
5. Transfer Money
6. View Transaction History
7. View Summary Statistics
8. Exit

Enter your choice:  4
Enter account number:  31058802008
Enter amount to withdraw:  10000

Withdrew 10000.0 successfully! New Balance: 792000.0

Bank Account Management System
1. Open a New Account
2. View Account Details
3. Deposit Money
4. Withdraw Money
```

```
5. Transfer Money
6. View Transaction History
7. View Summary Statistics
8. Exit

Enter your choice:  5
Enter sender account number:  31058802008
Enter recipient account number:  54778603772
Enter amount to transfer:  800

Transferred 800.0 from 31058802008 to 54778603772 successfully!

Bank Account Management System
1. Open a New Account
2. View Account Details
3. Deposit Money
4. Withdraw Money
5. Transfer Money
6. View Transaction History
7. View Summary Statistics
8. Exit

Enter your choice:  6
Enter account number:  54778603772

Date: 2024-12-07 17:00:32, Type: Transfer In, Amount: 800.0, Target
Account: 31058802008

Bank Account Management System
1. Open a New Account
2. View Account Details
3. Deposit Money
4. Withdraw Money
5. Transfer Money
6. View Transaction History
7. View Summary Statistics
8. Exit

Enter your choice:  7
Enter account number:  54778603772

Total Deposits: 0
Total Withdrawals: 0
Average Transaction Amount: 800.0

Bank Account Management System
1. Open a New Account
2. View Account Details
3. Deposit Money
4. Withdraw Money
5. Transfer Money
```

```
6. View Transaction History
7. View Summary Statistics
8. Exit

Enter your choice:  8

Exiting program. Goodbye!
```