Python Bank Account Management System

```
import numpy as np
import datetime as dt
import random
import pickle

user_details = {}
account_details = {}
credentials = {}
transactions = {}

try:
    with open("user_details.pkl", "rb") as f:
        user_details = pickle.load(f)
    with open("account_details.pkl", "rb") as f:
        account_details = pickle.load(f)
    with open("credentials.pkl", "rb") as f:
        credentials = pickle.load(f)
    with open("transactions.pkl", "rb") as f:
        transactions = pickle.load(f)
except (FileNotFoundError, EOFError):
    print("No existing data found.\n")

print("-----Welcome to Bank Account Management System-----")

while True:
    print("\n1. Open a new account")
    print("2. View account details")
    print("3. Perform a transaction (Deposit, Withdraw, Transfer)")
    print("4. View transaction history")
    print("5. Exit\n")

    choice = input("Enter your choice (1-5) :")
```

```python
if choice == "1":
    validate_name = lambda name: name.isalpha() and ' ' not in name
    while True:
        f_name = input("\nEnter account holder's first name : ").title()

        if not validate_name(f_name):
            print("Please enter a valid first name containing only alphabets and no spaces.")
        else:
            break

    while True:
        l_name = input("Enter account holder's last name : ").title()
        if not validate_name(l_name):
            print("Please enter a valid last name containing only alphabets and no spaces.")
        else:
            break

    print(f"\nHi {f_name} {l_name}, Welcome.\n")

    while True:
        account_type = input("Enter the type of account you want to open (savings/current) : ").title()
        if account_type == "Savings" or account_type == "Current":
            break
        else:
            print("You have entered the incorrect option\n")

    while True:
        try:
            balance = float(input(f"\n(Minimum amount required to open a {account_type} Account is 500)\nHow much amount would you like to deposit?"))
            if balance < 1000:
                print(f"\nYou need to deposit atleast 1000 to open a {account_type} Account")
                continue
            else:
                break
        except ValueError:
            print("Please enter a valid numeric amount.")
```

```
while True:
    account_number = "222222"
    for i in range(1,5):
        temp = random.randint(0,9)
        account_number += str(temp)

    unique = True
    for i in user_details.values():
        if account_number == i[2]:
            unique = False
            break

    if unique == True:
        break

while True:
    username = f_name
    for i in range(1,3):
        temp = random.randint(0,9)
        username += str(temp)

    if username not in user_details:
        break

password = ""
for i in range(1,5):
    temp = random.randint(0,9)
    password += str(temp)

user_details[username] = [f_name, l_name, account_number, account_type, balance]
account_details[account_number] = [username]
credentials[username] = password

print("\nAccount opened successfully.")
print("-"*50)
print(f"Account Holder's Name: {f_name} {l_name}")
print(f"Account Number: {account_number}")
print(f"Account Type: {account_type}")
print(f"Balance: {balance}")
print("-"*50)
print("We have assigned you a username & password.")
print("-"*50)
print(f"Username: {username}")
print(f"Password: {password}")
print("-"*50)

with open("user_details.pkl", "wb") as f:
        pickle.dump(user_details, f)
with open("account_details.pkl", "wb") as f:
    pickle.dump(account_details, f)
with open("credentials.pkl", "wb") as f:
    pickle.dump(credentials, f)
```

```
elif choice == "2":
    print("Login to your account")

    while True:
        username_input = input("Enter your username : ")
        if username_input not in credentials:
            print("Username doesn't exist. Please enter a valid username.")
            continue
        else:
            break

    while True:
        password_input = input("Enter your password : ")
        if password_input != credentials[username_input]:
            print("Password is incorrect. Please enter the correct password.")
        else:
            print("\nLogin Successful.")
            break

    print("-"*50)
    print(f"Account Holder's Name : {user_details[username_input][0]} {user_details[username_input][1]}")
    print(f"Account Number        : {user_details[username_input][2]}")
    print(f"Account Type          : {user_details[username_input][-2]}")
    print(f"Balance               : {user_details[username_input][-1]}")
    print("-"*50)

elif choice == "3":
    print("Login to your account")

    while True:
        username_input = input("Enter your username : ")
        if username_input not in credentials:
            print("Username doesn't exist. Please enter a valid username.")
            continue
        else:
            break

    while True:
        password_input = input("Enter your password : ")
        if password_input != credentials[username_input]:
            print("Password is incorrect. Please enter the correct password.")
        else:
            print("\nLogin Successful.")
            break

    if username_input not in transactions.keys():
        transactions[username_input] = {}

    print("\n1. Deposit")
    print("2. Withdraw")
    print("3. Transfer")
```

```python
while True:
    option = input("Enter your choice (1-3) :")
    type_of_transaction = ""

    if option == "1":
        type_of_transaction = "Deposit"
        while True:
            try:
                deposit_amount = float(input(f"\n(Minimum deposit amount : 500)\nHow much amount w
                if deposit_amount < 500:
                    print(f"\nInvalid deposit amount. You need to deposit atleast 500")
                    continue
                else:
                    break
            except ValueError:
                print("Please enter a valid numeric amount.")

        user_details[username_input][-1] += deposit_amount
        current_time = dt.datetime.now()
        print("\nDeposit Successful.")
        print("-"*50)
        print(f"Amount Deposited: {deposit_amount}")
        print(f"Updated Balance: {user_details[username_input][-1]}")
        print("-"*50)
```

```python
while True:
    option = input("Enter your choice (1-3) :")
    type_of_transaction = ""

    if option == "1":
        type_of_transaction = "Deposit"
        while True:
            try:
                deposit_amount = float(input(f"\n(Minimum deposit amount : 500)\nHow much amount would you like to deposit?"))
                if deposit_amount < 500:
                    print(f"\nInvalid deposit amount. You need to deposit atleast 500")
                    continue
                else:
                    break
            except ValueError:
                print("Please enter a valid numeric amount.")

        user_details[username_input][-1] += deposit_amount
        current_time = dt.datetime.now()
        print("\nDeposit Successful.")
        print("-"*50)
        print(f"Amount Deposited: {deposit_amount}")
        print(f"Updated Balance: {user_details[username_input][-1]}")
        print("-"*50)


        transactions[username_input][current_time] = [type_of_transaction, deposit_amount, user_details[username_input][-1]]

        break

    elif option == "2":
        if user_details[username_input][-1] > 0:
            type_of_transaction = "Withdraw"
            while True:
                try:
                    withdraw_amount = float(input("\nHow much amount would you like to withdraw?"))
                    if withdraw_amount < 0:
                        print("\nAmount to be withdrawn should be higher than 0.")
                        continue
                    elif withdraw_amount > user_details[username_input][-1]:
                        print(f"\nYou don't have enough balance to withdraw. Available Balance : {user_details[username_input][-1]}")
                        continue
                    else:
                        break
                except ValueError:
                    print("Please enter a valid numeric amount.")

            user_details[username_input][-1] -= withdraw_amount
            current_time = dt.datetime.now()
            print("\nWithdraw Successful.")
            print("-"*50)
            print(f"Amount Withdrawn: {withdraw_amount}")
            print(f"Updated Balance: {user_details[username_input][-1]}")
            print("-"*50)
```

```
        transactions[username_input][current_time] = [type_of_transaction, withdraw_amount, user_details[username_input][-1]]


        break

    else:
        print(f"You don't have enough balance to withdraw (Available Balance : {user_details[username_input][-1]}).")


elif option == "3":
    if user_details[username_input][-1] > 0:
        type_of_transaction = "Transfer"
        while True:
            transfer_account = input("Please enter the account number to which you would like to transfer the amount :")
            acc_found = False
            same_account = False

            for i in user_details.values():
                if transfer_account != user_details[username_input][2]:
                    if transfer_account == i[2]:
                        acc_found = True
                        break
                else:
                    same_account = True
                    break

            if acc_found and not same_account:
                while True:
                    try:
                        transfer_amount = float(input("Please enter the amount you would like to transfer : "))
                        if transfer_amount > user_details[username_input][-1]:
                            print(f"\nYou don't have enough balance to transfer. Available Balance : {user_details[username_input][-1]}")
                            continue

                        elif transfer_amount < 0:
                            print("\nYou cannot transfer an amount lower than 0.")
                            continue

                        else:
                            user_details[username_input][-1] -= transfer_amount
                            sender_balance = user_details[username_input][-1]

                            recipient_user = next(user for user, details in user_details.items() if details[2] == transfer_account)

                            user_details[recipient_user][-1] += transfer_amount
                            recipient_balance = user_details[recipient_user][-1]

                            current_time = dt.datetime.now()
```

```
                                    print("\nTransfer Successful.")
                                    print("-" * 50)
                                    print(f"Amount transferred : {transfer_amount}")
                                    print("-" * 50)
                                    print(f"Account transferred from : {user_details[username_input][2]}")
                                    print(f"Updated Balance : {sender_balance}")
                                    print("-" * 50)
                                    print(f"Account transferred to : {transfer_account}")
                                    print(f"Updated Balance : {recipient_balance}")
                                    print("-" * 50)

                                    if username_input not in transactions:
                                        transactions[username_input] = {}
                                    transactions[username_input][current_time] = [f"{type_of_transaction}(Sent)", transfer_amount, sender_balance]

                                    if recipient_user not in transactions:
                                        transactions[recipient_user] = {}
                                    transactions[recipient_user][current_time] = [f"{type_of_transaction}(Received)", transfer_amount, recipient_b

                                    break

                            except ValueError:
                                print("\nPlease enter a valid numeric amount.")
                        break

                    elif same_account:
                        print("\nYou cannot transfer money to your own account.")

                    else:
                        print("\nInvalid account number. Please try again.")

                break

            else:
                print(f"\nYou don't have enough balance to transfer (Available Balance : {user_details[username_input][-1]}). Please deposit some

        else:
            print("\nInvalid choice, please try again")

    with open("transactions.pkl", "wb") as f:
        pickle.dump(transactions, f)

elif choice == "4":

    while True:
        username_input = input("Enter the username to view the transactions : ")
        if username_input in transactions:
```

```
    print("-" * 80)
    print(f"User: {username_input}")
    print(f"Account No: {user_details[username_input][2]}")
    print("-" * 80)
    print(f"{'Date & Time':<28} {'Transaction':<22} {'Amount':<15} {'Balance':<11}")
    print("-" * 80)

    for date_time, (transaction_type, amount, balance) in transactions[username_input].items():
        print(f"{str(date_time):<28} {transaction_type:<22} {amount:<15.2f} {balance:<11.2f}")
    print("-" * 80)

    user_transactions = transactions[username_input]

    deposits = []
    withdrawals = []
    transfers_sent = []
    transfers_received = []

    for transaction_details in user_transactions.values():
        transaction_type, amount, _ = transaction_details

        if "Deposit" in transaction_type:
            deposits.append(amount)
        elif "Withdraw" in transaction_type:
            withdrawals.append(amount)
        elif "Transfer(Sent)" in transaction_type:
            transfers_sent.append(amount)
        elif "Transfer(Received)" in transaction_type:
            transfers_received.append(amount)

    deposits = np.array(deposits)
    withdrawals = np.array(withdrawals)
    transfers_sent = np.array(transfers_sent)
    transfers_received = np.array(transfers_received)

    total_deposits = np.sum(deposits) if deposits.size > 0 else 0
    total_withdrawals = np.sum(withdrawals) if withdrawals.size > 0 else 0
    total_transfers_sent = np.sum(transfers_sent) if transfers_sent.size > 0 else 0
    total_transfers_received = np.sum(transfers_received) if transfers_received.size > 0 else 0

    all_transactions = np.concatenate((deposits, withdrawals, transfers_sent, transfers_received))
    average_transaction_amount = np.mean(all_transactions) if all_transactions.size > 0 else 0

    print(f"\nSummary for {username_input}:")
    print(f"Total Deposits: {total_deposits:.2f}")
    print(f"Total Withdrawals: {total_withdrawals:.2f}")
    print(f"Total Amount Sent: {total_transfers_sent:.2f}")
    print(f"Total Amount Received: {total_transfers_received:.2f}")
    print(f"Average Transaction Amount: {average_transaction_amount:.2f}\n")
    print("-" * 80)
```

```
-----Welcome to Bank Account Management System-----

1. Open a new account
2. View account details
3. Perform a transaction (Deposit, Withdraw, Transfer)
4. View transaction history
5. Exit

Enter your choice (1-5) : 1

Enter account holder's first name :  bhu5
Please enter a valid first name containing only alphabets and no spaces.

Enter account holder's first name :  Bhuvan
Enter account holder's last name :  Jari

Hi Bhuvan Jari, Welcome.

Enter the type of account you want to open (savings/current) :  savings

(Minimum amount required to open a Savings Account is 500)
How much amount would you like to deposit? 10,000
Please enter a valid numeric amount.

(Minimum amount required to open a Savings Account is 500)
How much amount would you like to deposit? 10000

Account opened successfully.
--------------------------------------------------
Account Holder's Name: Bhuvan Jari
Account Number: 2222220164
Account Type: Savings
Balance: 10000.0
--------------------------------------------------
We have assigned you a username & password.
--------------------------------------------------
Username: Bhuvan53
Password: 1489
--------------------------------------------------
```

```
1. Open a new account
2. View account details
3. Perform a transaction (Deposit, Withdraw, Transfer)
4. View transaction history
5. Exit

Enter your choice (1-5) : 2
Login to your account
Enter your username :  Bhuvan53
Enter your password :  1489

Login Successful.
--------------------------------------------------
Account Holder's Name : Bhuvan Jari
Account Number        : 2222220164
Account Type          : Savings
Balance               : 10000.0
--------------------------------------------------

1. Open a new account
2. View account details
3. Perform a transaction (Deposit, Withdraw, Transfer)
4. View transaction history
5. Exit

Enter your choice (1-5) : 3
Login to your account
Enter your username :  Bhuvan53
Enter your password :  1489

Login Successful.

1. Deposit
2. Withdraw
3. Transfer
Enter your choice (1-3) : 2

How much amount would you like to withdraw? 5000

Withdraw Successful.
--------------------------------------------------
Amount Withdrawn: 5000.0
Updated Balance: 5000.0
--------------------------------------------------
```

```
1. Open a new account
2. View account details
3. Perform a transaction (Deposit, Withdraw, Transfer)
4. View transaction history
5. Exit

Enter your choice (1-5) : 4
Enter the username to view the transactions :  Bhuvan53
--------------------------------------------------------------------------------
User: Bhuvan53
Account No: 2222220164
--------------------------------------------------------------------------------
Date & Time                  Transaction            Amount          Balance
--------------------------------------------------------------------------------
2025-03-05 16:03:22.680002   Withdraw               5000.00         5000.00
--------------------------------------------------------------------------------

Summary for Bhuvan53:
Total Deposits: 0.00
Total Withdrawals: 5000.00
Total Amount Sent: 0.00
Total Amount Received: 0.00
Average Transaction Amount: 5000.00

--------------------------------------------------------------------------------

1. Open a new account
2. View account details
3. Perform a transaction (Deposit, Withdraw, Transfer)
4. View transaction history
5. Exit

Enter your choice (1-5) : 5

Exit successful.
```