In [ ]:
```python
from datetime import datetime

# Data storage
db = {"accounts": {}, "transactions": {}}

def generate_account_number():
    return str(100000 + len(db["accounts"]))

def create_account():
    name = input("Enter account holder name: ")
    account_type = input("Enter account type (savings/current): ").lower()
    initial_balance = float(input("Enter initial deposit amount: "))
    account_number = generate_account_number()
    db["accounts"][account_number] = {
        "name": name,
        "account_type": account_type,
        "balance": initial_balance
    }
    print(f"Account created successfully! Account Number: {account_number}")

def view_account():
    account_number = input("Enter account number: ")
    if account_number in db["accounts"]:
        account = db["accounts"][account_number]
        print(f"Name: {account['name']}, Account Type: {account['account_type']}, Balance: {account['balance']}")
    else:
        print("Account not found!")

def deposit():
    account_number = input("Enter account number: ")
    if account_number in db["accounts"]:
        amount = float(input("Enter deposit amount: "))
        db["accounts"][account_number]["balance"] += amount
        db["transactions"].setdefault(account_number, []).append((datetime.now(), "Deposit", amount))
        print("Deposit successful!")
    else:
        print("Account not found!")

def withdraw():
    account_number = input("Enter account number: ")
    if account_number in db["accounts"]:
        amount = float(input("Enter withdrawal amount: "))
        if amount > db["accounts"][account_number]["balance"]:
            print("Insufficient balance!")
        else:
            db["accounts"][account_number]["balance"] -= amount
            db["transactions"].setdefault(account_number, []).append((datetime.now(), "Withdrawal", amount))
            print("Withdrawal successful!")
    else:
        print("Account not found!")

def transfer():
    from_acc = input("Enter your account number: ")
    to_acc = input("Enter recipient account number: ")
    if from_acc in db["accounts"] and to_acc in db["accounts"]:
        amount = float(input("Enter transfer amount: "))
        if amount > db["accounts"][from_acc]["balance"]:
            print("Insufficient balance!")
        else:
            db["accounts"][from_acc]["balance"] -= amount
            db["accounts"][to_acc]["balance"] += amount
            db["transactions"].setdefault(from_acc, []).append((datetime.now(), "Transfer to " + to_acc, amount))
            db["transactions"].setdefault(to_acc, []).append((datetime.now(), "Transfer from " + from_acc, amount))
            print("Transfer successful!")
    else:
        print("Invalid account number!")

def view_transactions():
    account_number = input("Enter account number: ")
    if account_number in db["transactions"]:
        print("Transaction History:")
        for txn in db["transactions"][account_number]:
            print(f"Date: {txn[0]}, Type: {txn[1]}, Amount: {txn[2]}")
    else:
        print("No transactions found!")

def main():
    while True:
        print("\nBank Account Management System")
        print("1. Open a New Account")
        print("2. View Account Details")
        print("3. Deposit")
        print("4. Withdraw")
        print("5. Transfer")
        print("6. View Transaction History")
        print("7. Exit")

        choice = input("Enter your choice: ")

        if choice == "1":
            create_account()
        elif choice == "2":
            view_account()
        elif choice == "3":
            deposit()
        elif choice == "4":
            withdraw()
        elif choice == "5":
            transfer()
        elif choice == "6":
            view_transactions()
        elif choice == "7":
            print("Exiting the program...")
            break
        else:
            print("Invalid choice! Please try again.")

if __name__ == "__main__":
    main()
```

```
Bank Account Management System
1. Open a New Account
2. View Account Details
3. Deposit
4. Withdraw
5. Transfer
6. View Transaction History
7. Exit
Account created successfully! Account Number: 100000

Bank Account Management System
1. Open a New Account
2. View Account Details
3. Deposit
4. Withdraw
5. Transfer
6. View Transaction History
7. Exit
Account not found!

Bank Account Management System
1. Open a New Account
2. View Account Details
3. Deposit
4. Withdraw
5. Transfer
6. View Transaction History
7. Exit
Name: ansh, Account Type: saveings, Balance: 500000.0

Bank Account Management System
1. Open a New Account
2. View Account Details
3. Deposit
4. Withdraw
5. Transfer
6. View Transaction History
7. Exit
Deposit successful!

Bank Account Management System
1. Open a New Account
2. View Account Details
3. Deposit
4. Withdraw
5. Transfer
6. View Transaction History
7. Exit
Withdrawal successful!

Bank Account Management System
1. Open a New Account
2. View Account Details
3. Deposit
4. Withdraw
5. Transfer
6. View Transaction History
7. Exit
Invalid account number!

Bank Account Management System
1. Open a New Account
2. View Account Details
3. Deposit
4. Withdraw
5. Transfer
6. View Transaction History
7. Exit
No transactions found!

Bank Account Management System
1. Open a New Account
2. View Account Details
3. Deposit
4. Withdraw
5. Transfer
6. View Transaction History
7. Exit
Transaction History:
Date: 2025-02-01 22:30:18.598265, Type: Deposit, Amount: 58000.0
Date: 2025-02-01 22:30:43.769393, Type: Withdrawal, Amount: 280000.0

Bank Account Management System
1. Open a New Account
2. View Account Details
3. Deposit
4. Withdraw
5. Transfer
6. View Transaction History
7. Exit
Name: ansh, Account Type: saveings, Balance: 278000.0

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

6. View Transaction History
7. Exit

In [ ]: