

# Bank-App Jack Bravo OPA21MA

Github-repo: <https://github.com/bravojack/BankApp.git>

## Inledning:

### A. Beskrivning av projektet och målet:

Jag skall skapa ett system för en fiktiv bank. Banken skall ha ett antal kunder, och varje kund ett eller flera bank-konton.

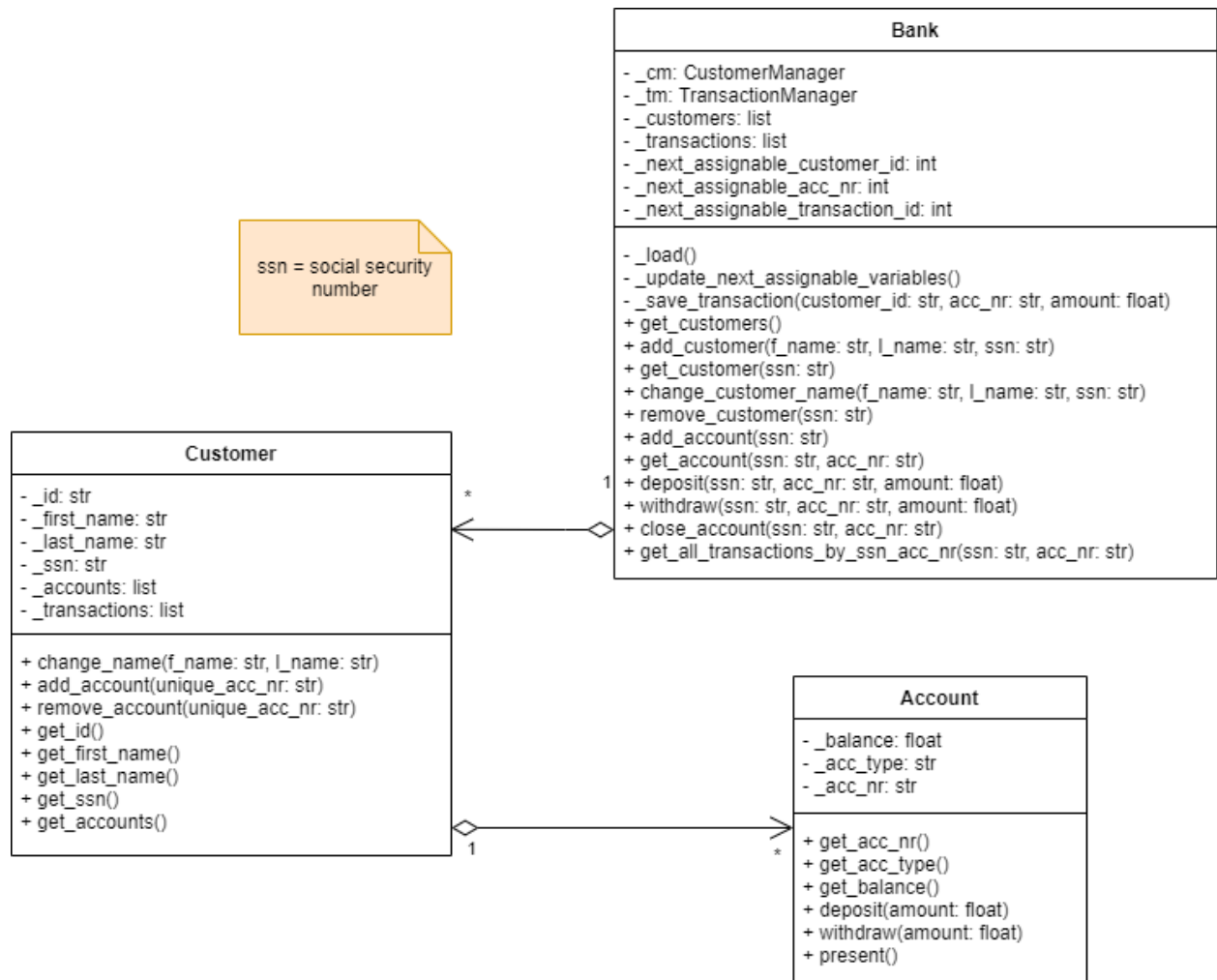
### B. User Stories:

- Som användare vill jag kunna se en lista med alla kunder i banken
- Som användare vill kunna se information om en kund inklusive deras konton
- Som användare vill jag kunna lägga till och ta bort kunder
- Som användare vill jag kunna skapa och ta bort konton som tillhör kunder
- Som användare vill jag kunna sätta in pengar på ett konto
- Som användare vill jag kunna ta ut pengar från ett konto, men bara om det finns tillräckligt på saldot.
- Som användare vill jag kunna se alla transaktioner som en kund gjort med ett visst konto

### C. Teknologier som används:

- Python
- Datetime

## Klassdiagram:



## **Milestones:**

### **1. Skapa klass-diagram**

Beskrivning: Skapa ett klass-diagram för klasserna Bank, Customer och Account

Tasks:

- Skapa klass-diagrammet

### **2. Implementera klasser**

Beskrivning: Jag börjar med att skapa alla klasser, med funktionalitet så som jag tror att de behövs utformas enligt instruktioner.

Tasks:

- Implementera klassen Account
- Implementera klassen Customer
- Implementera klassen Bank
- Testa funktionalitet för existerande klasser

### **3. Implementera funktionalitet för att läsa från fil**

Beskrivning: Jag implementerar klassen DataSource och skapar mock-data i en fil. Jag testar så att metoderna fungerar som de skall.

Tasks:

- Skapa mockdata i en textfil
- Implementera klassen DataSource
- Testa funktionalitet för DataSource

### **4. Koppla samman DataSource-klassen med övriga klasser**

Beskrivning: Jag kopplar samman klasserna och slutför funktionalitets-implementering. Jag testar så att applikationen fungerar i enlighet med önskemål.

Tasks:

- Se till så att alla klasser är korrekt kopplade, inklusive DataSource

### **5. Skapa ett user interface**

Beskrivning: Jag skapar ett menysystem för en console-applikation. Jag skapar en egen klass som heter UI som sköter allt relaterat till user interface

- Skapa och implementera UI klassen
- Testa appen och se till så att menysystemet ser snyggt ut

## **6. Sluttest och avrundning av projekt**

- Slut-testa Applikationen
- Töm data och fyll den med snygg mock-data
- Skriv ner avslutande tankar kring projektet
- Ladda upp till github repo

### **Avslutande tankar:**

Jag tycker detta har varit ett roligt och lärorikt projekt. Jag har fått googla en hel del för att komma in i hur python fungerar, men tack vare det har det satt sig ganska bra nu. Jag tycker att ordningen jag valde att göra projektet fungerade bra. Det enda jag inte förstår är hur tanken är att vi skall använda 'datasource\_conn' metoden i DataSource-klassen. Jag antar att den egentligen skall kallas när man startar appen för att se så att den har en connection till filen som skall läsas och skrivas från. Jag har inte haft några större problem med denna uppgiften generellt sett, utan allt har gått att lösa med lite googling eller debugging.