Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

UC Sistemas de Informação e Bases de Dados

Bank System

Professor João Silva

Ana Rita de Almeida Martinho, up201709727

Gonçalo António de Jesus Xavier, up201604506

Group 11

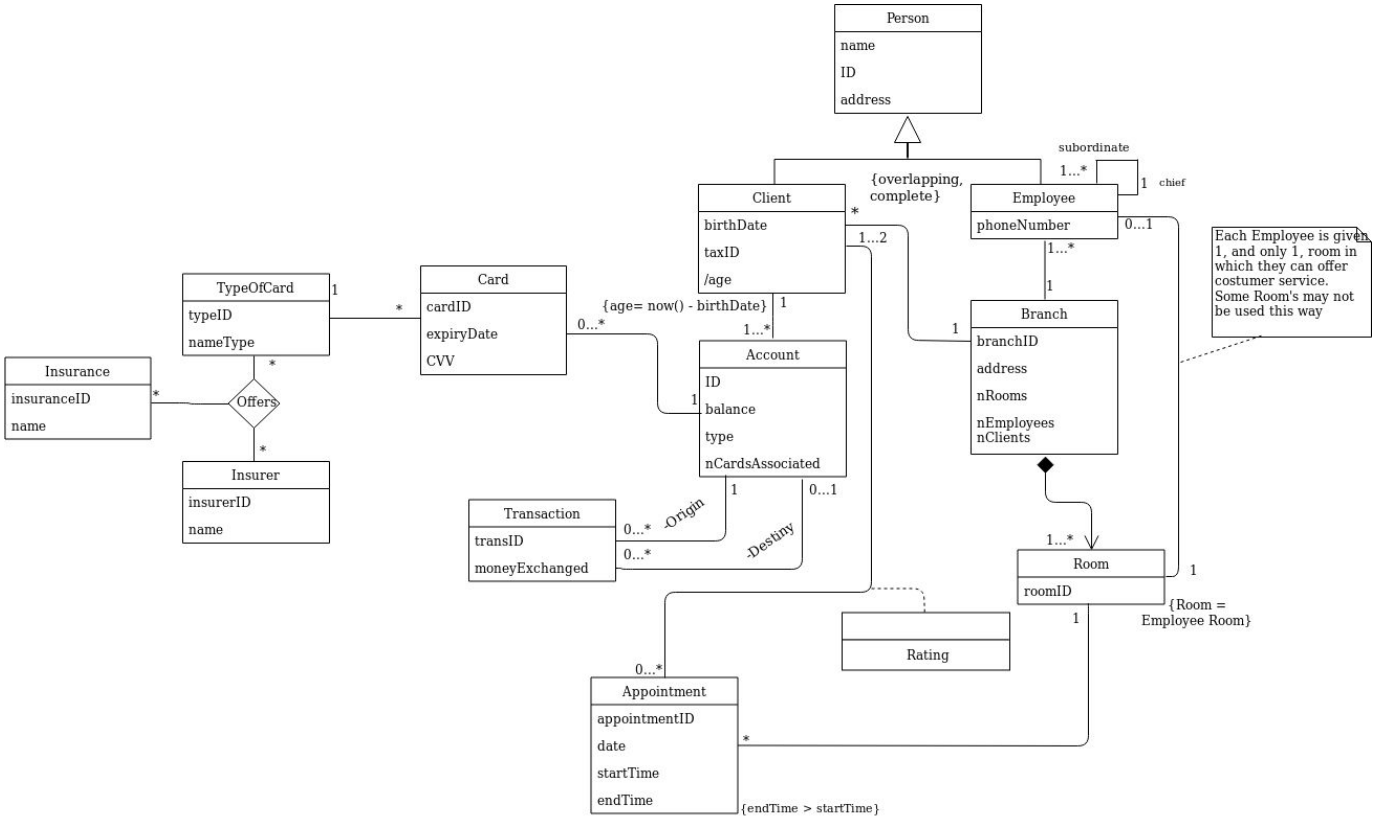
Novembro 2019

Description

The owner of a bank wants to create a database to store information about all the the services the bank performs. After some research the owner came up with the following requirements:

- For each client, its name, id (that must be unique), address, date of birth, age (for campaign purposes) and taxID must be stored;
- The bank has several employees. For each one, the name, id (unique), address and phone number must be stored;
- Each client, has one or many accounts associated. A client cannot have no account because that goes against the definition of "client".
- An account has the following attributes: id (number of the account), client, balance, type (current or term), number of cards associated.
- Each account can have many cards. For each card, these informations must be in the data-base: number of the card (unique), expiry date, CVV, and type (debit or credit).
- The bank, of course, also has branches spread around the country. Each branch has an id, address, number of employees, clients and rooms.
- Each branch has one employee that is the chief of the branch and all the other ones are subordinates.
- A client is linked to one and only branch. A branch can have lots of clients.
- The employees can provide customer service.
- A client can go to the physical location of it's branch and have an appointment with an employee (this can happen whenever a client has a problem he wants to address or wants to perform a some form of a special bank operation).
- Each appointment can have 1 or 2 clients (2 clients can go and perform some special operation together, such as a joint-loan), a unique ID, an employee associated, a date and a start and end hour.
- The branch has several rooms, each employee has a room assigned.
- All the appointments occur in the employee’s room .
- The owner of the bank wants to know the quality of the bank’s appointments, therefore, every client must rate the appointment (scale: 0 to 10).
- The clients (through the account) can exchange money between them immediately, i.e a transaction.
- A transaction has an unique ID,the money exchanged, the origin account and the destiny account.
- A client can do a transaction with no destiny account, ie, a payment.
- The bank has partnerships with several insurers, and, depending on the type of the card, the insurers can offer different types of insurances (more offers if it’s a credit card,e.g).

UML Class Diagram



Relational Model

- Person(ID, name(NOT NULL), address){overlapping, complete}

- Client(ID->Person, birthDate, taxID, /age, branch_ID->Branch)
  {age= now() - birthDate}

- Employee(ID->Person, phoneNumber, branch_ID -> Branch (NOT NULL), room_ID->room)

- Relationship(chief->Employee, subordinate->Employee)

  Account(Account_ID, balance, type , nCardsAssociated, client_ID->Client (NOT NULL))

- Transaction(trans_ID, moneyExchanged, Origin_Account_ID -> Account (NOT NULL), Destiny_Account_ID -> Account)

- Card(card_ID, expiryDate, CVV, associated_account->Account (NOT NULL), type_of_card->TypeOfCard)

- TypeOfCard(typeID, nameType)

- Insurance(insurance_ID, name)

- Insurer(insurer_ID, name)

- Offers(insurer_ID->Insurer, insurance_ID->Insurance, typeID->TypeOfCard)

- Appointment(appointment_ID, date, startTime, endTime, ID->Client, room_ID->Room)
  {endTime > startTime}

- Rating(ID->Client, appointment->Appointment, rating_score)

- Room(room_ID, branch_ID->Branch (NOT NULL))

- Branch(branch_ID, address, nEmployees, nClients, nRooms)