Zihao Zhang
Huanhua Xu
CS174 Project report

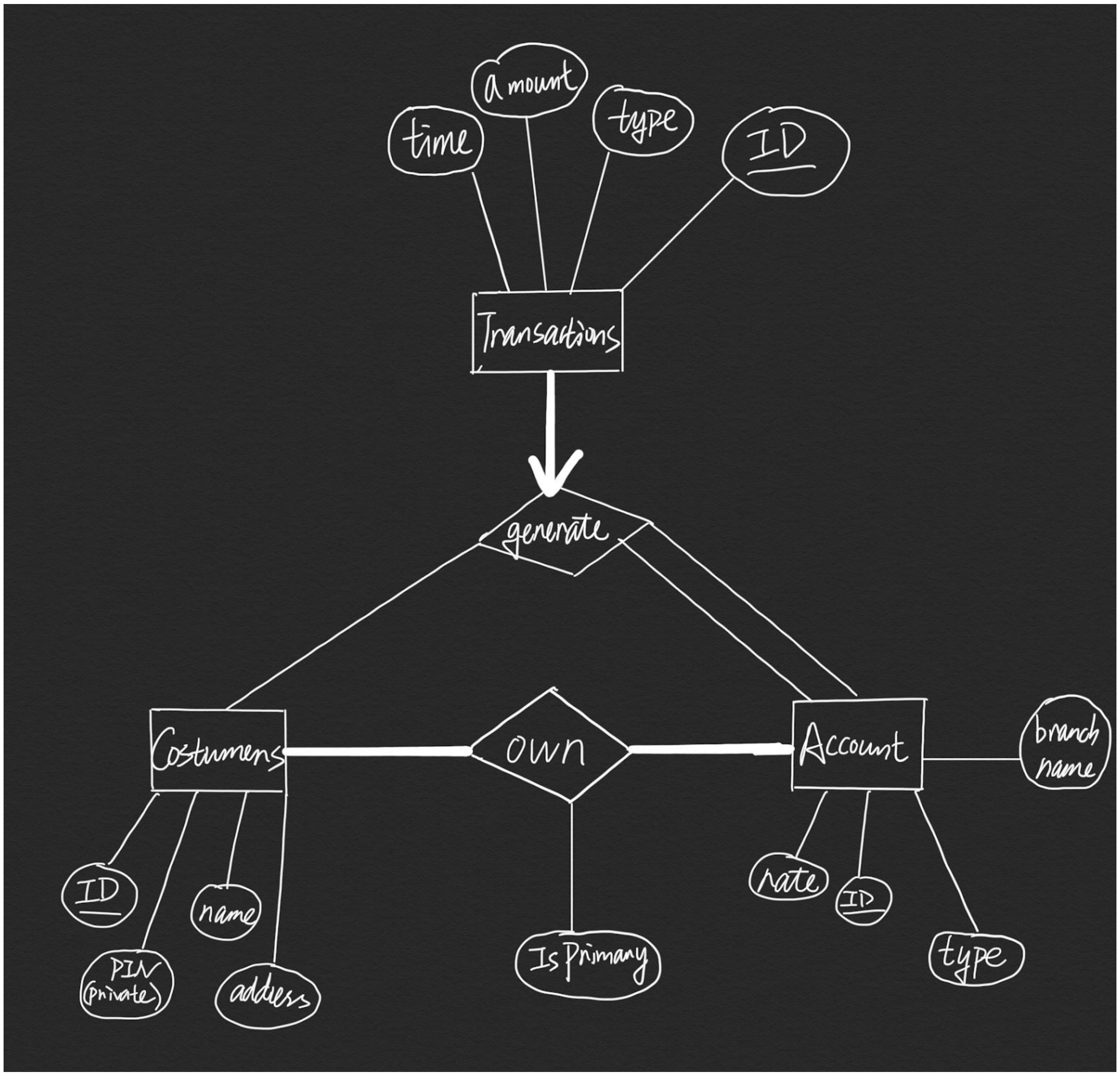
1. Identify as many integrity constraints as you can on your initial ER diagram. You may describe the constraints in English.

    1.1. Account:
        1.1.1. An account can have more than one owner, but only one of them is designated as the *primary* owner;
        1.1.2. Each account has uniquely identified by account ID number (an integer), a list of transactions made during the month, and a bank branch name.
        1.1.3. Accounts come in three classes: checking, savings, and pocket;
        1.1.4. When a checking or savings account is first created, it must have a balance of at least $1,000;
        1.1.5. When a pocket account is created the customer must already have a checking or savings account with a positive balance;
        1.1.6. No transaction can make an account balance to go below $0.00;
        1.1.7. There are two types of checking account: student checking and interest checking;
        1.1.8. At the end of each month, all open account earn interest on their balance. Interest check has initial 3.00% annual rate; saving account has initial 4.80% interest; student checking and pocket account earns no interest;
        1.1.9. All successful transactions on an account should be recorded for the account;

    1.2. Costumer:
        1.2.1. A customer is an individual with a name, an address, and a (unique) tax identification number;
        1.2.2. Every customer has a set of (jointly) owned accounts;
        1.2.3. Each customer has a unique PIN. The PIN should be private data;

    1.3. Transaction:
        1.3.1. The following transaction types are allowed in the system: Deposit, Top-Up, Withdrawal, Purchase, Transer, Collect, Pay-Friend, Wire, Write-Check, Accrue-Interest
        1.3.2. Associated with every transaction is the date of the transaction and the account(s) involved;

2. Design an ER diagram for the application described in the project and express as many integrity con- straints you have identified as possible.



3. Translate the ER diagram into relation schemas and do not forget the integrity constraints you have identified.

   Accounts(account_id: string, branch_name: string, account_type: string, rate: real)

   Customers(tax_id: string, name: string, address: string, PIN: integer)

   Transactions(transaction_id: string, type: string, time: datetime, amount: real)

   Generate(tax_id: string, transaction_id: string, account_id_one: string, account_id_two: string)

   Own(tax_id: string, account_id: string, isprimary: boolean)

4. Indicate which integrity constraints that your relational database schema is able to incorporate; identify additional integrity constraints if possible.

   Key constraint and Total participation constraint in the Ternary ‘generate’ relationship: Every transaction is unique and every transaction participates in ‘generate’ only once.

   Primary key constraint: Accounts has primary key ‘account_id’, Customers has primary key ‘tax_id, Transactions has primary key ‘transaction_id.

Generate is identified by a combination primary key of (tax_id, transaction_id, account_id_one, account_id_two), in which ‘tax_id’ is foreign key reference to Customers, ‘account_id_one’ and ‘account_id_two’ are foreign keys reference to Accounts, ‘transaction_id’ is foreign key reference to Transactions.

Own is identified by a combination primary key of (tax_id: string, account_id: string), where ‘tax_id’ is a foreign key reference to Customers, ‘account_id’ is a foreign key reference to Accounts.

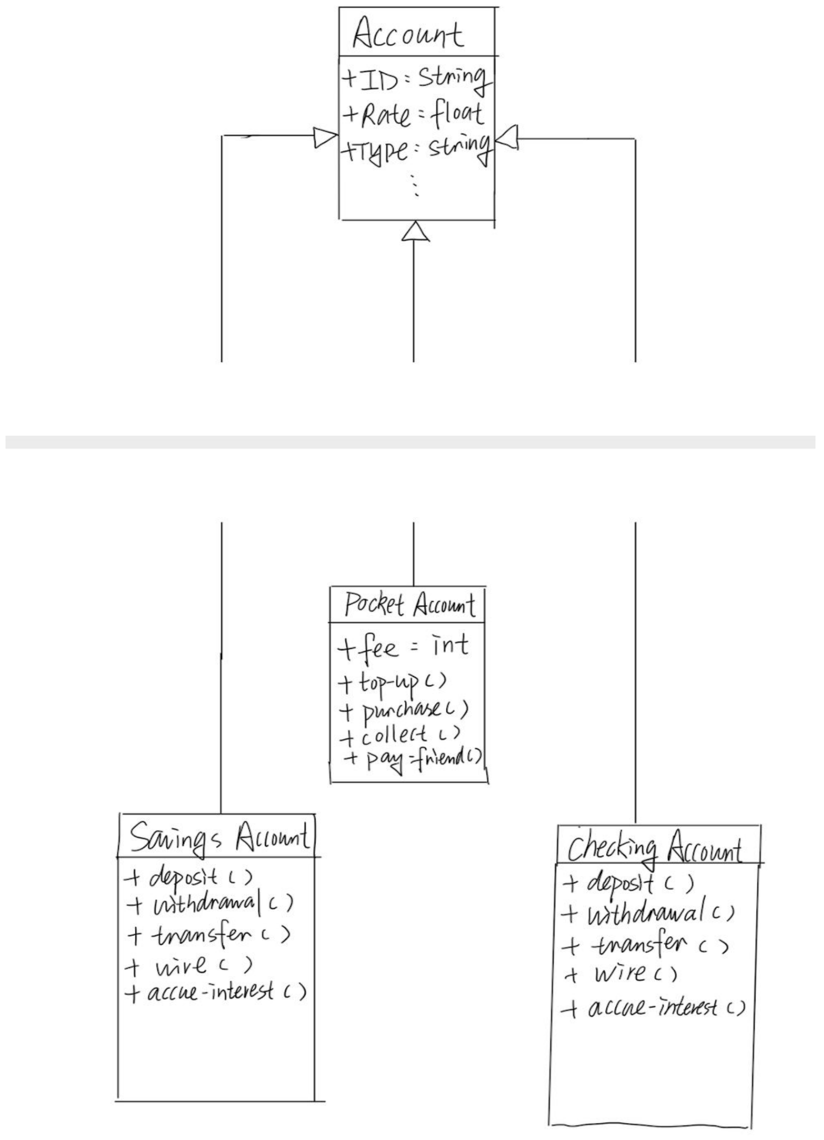
5. Discuss briefly how you will deal with a violation of each of the integrity constraints identified.

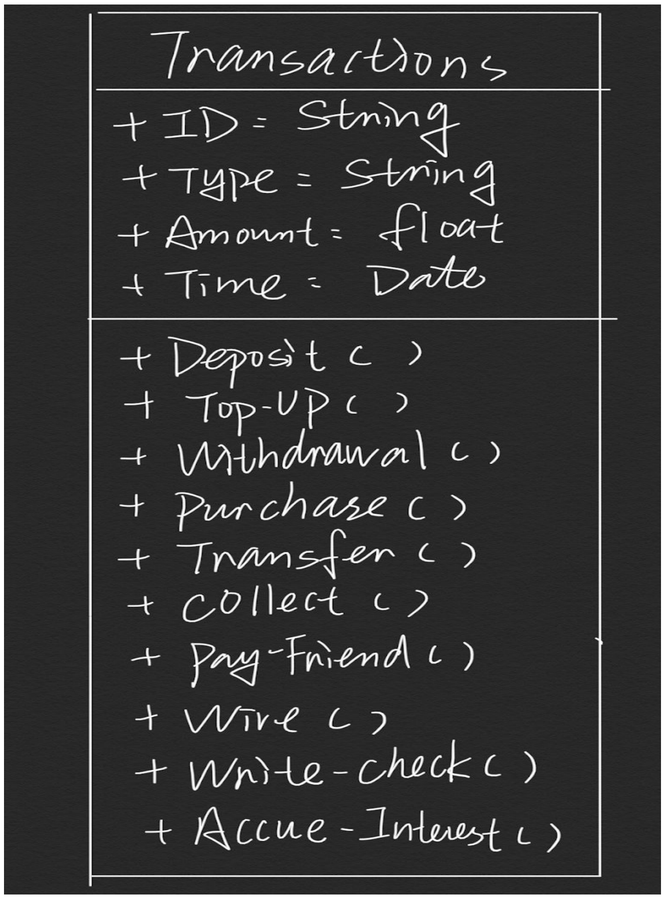
   Primary keys can not be NULL in Entity Accounts, Customers, Transactions.

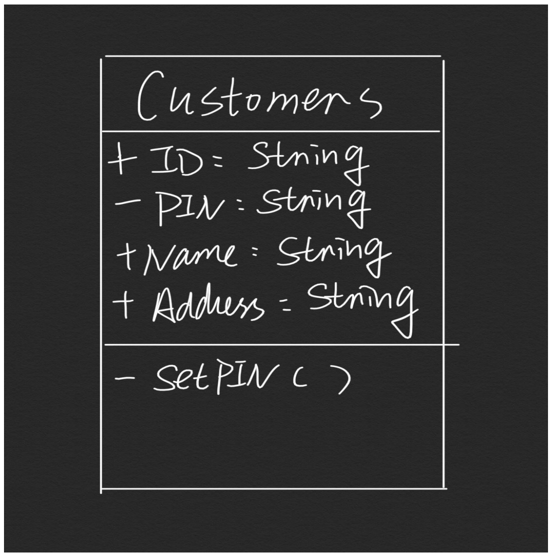
In relationship Generate, account_id_two can be NULL if the transaction involves only one account (eg. deposit), and it is not NULL if the transaction involves two accounts. The  account_id_one can NOT be NULL at all time. The Generate relationship has a combined primary key (tax_id, transaction_id, account_id_one, account_id_two). We set Generate to be ON DELETE CASCADE.

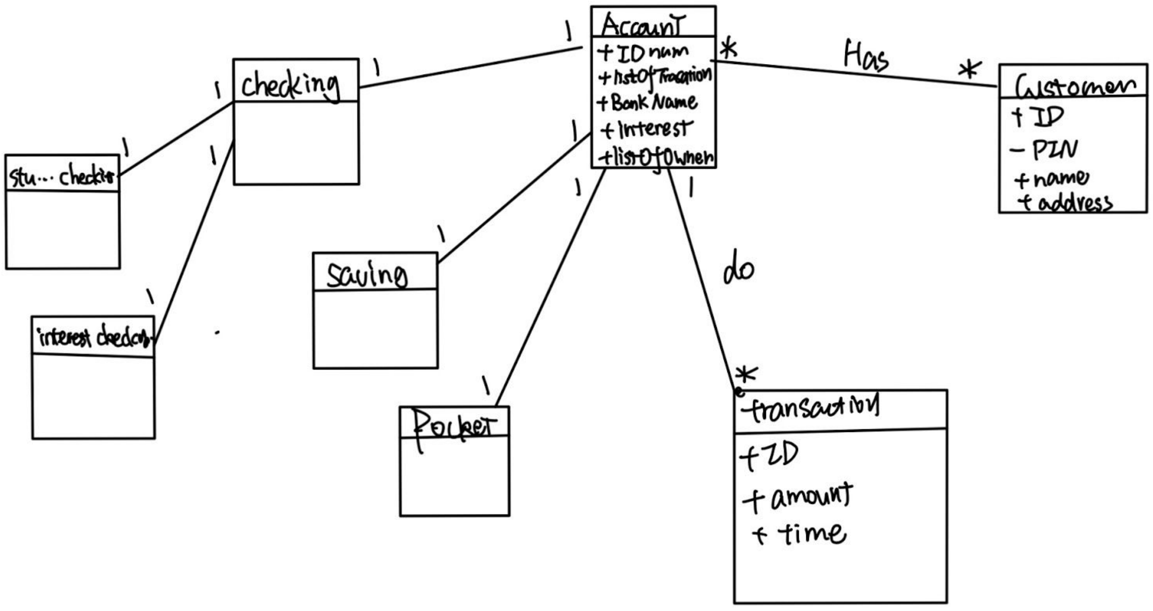
In relationship Own, neither tax_id and transaction_id can be NULL. They together uniquely identifies a Own relationship. We set Own to be ON DELETE CASCADE. When either the entry with tax_id or transaction_id is deleted, the corresponding entry is Own is deleted immediately as well.

6. Provide an initial system design. You can choose appropriate means to describe your system design, e.g., class diagrams, diagram of functional components, etc.




Transactions
+ ID : String
+ Type : String
+ Amount : float
+ Time : Date
+ Deposit ( )
+ Top-Up ( )
+ Withdrawal ( )
+ Purchase ( )
+ Transfer ( )
+ Collect ( )
+ Pay-Friend ( )
+ Wire ( )
+ Write-check ( )
+ Accue-Interest ( )


Customers
+ ID : String
− PIN : String
+ Name : String
+ Addess : String
− SetPIN ( )



7. List the task divisions and list each member's responsibility.

JDBC & project set up, Java Class Models construction; banker functions implementation, customer class implementation, debug

Front end GUI interfaces ( Customer & Banker ), Java Class Models construction, customer functions implementation,debug