NJIT | CS 631 | DATA MANAGEMENT SYSTEMS DESIGN
Fall 2018

PROJECT

Purpose of this project

Analyze, design, implement, and document a database system application. You will use the methodology for database development learned in class. The system must be implemented on a DBMS with any language as a host-language for the application (use of Java and accessing the database using JDBC is one possibility). The system can be implemented as a desktop, web, or mobile application and must have a Graphical User Interface that includes the basic functionality described below.

The TIJN Payment Network

The following specifications are intended as a guide; they are not the complete specifications. These are intended to be a basis for you to get started in the right direction in designing your system. You as the designer must analyze and decide what other details or features should be specified for your system. Thus, individual group implementations will differ in terms of design and implementation styles. Every group has to mention clearly in its report what other specifications are assumed.

Data and Functional Requirements

TIJN is a payment network, similar to Venmo and Zelle, that enables individuals to electronically transfer money to others.

1. Users can sign up with TIJN (that is, create an account with TIJN) by providing their name, SSN, an email address, and a phone number. Email addresses and phone numbers should be verified – e.g. a user can prove that owns an email address or a phone number by entering correctly the code that is sent via email or SMS respectively. Bank accounts are identified by the bank ID and the account number.

2. Users can link multiple bank accounts with their account but they should define one of them as the primary funding source. Users can verify a bank account by inserting the very small amount that TIJN deposited to that account. Users are also able to unlink a bank account.

3. There are two forms of payment that TIJN supports:

    a. Send money to a user
    b. Request money from users

4. **Send money:** This type of payment can by done by specifying the recipient's email address or phone number, the amount, and a memo (optional). If the amount is over the sender's TIJN balance, the system will fund the entire payment from the sender's primary funding source.

5. **Request money:** A user can request money from others by specifying their email addresses or phone numbers, the amounts, and a memo (optional). The typical use case could be to "Split a bill", i.e. a charge can be divided among friends and family equally, or with different amounts, together adding up to the bill total.

6. A payment to an email address or phone number that isn't associated with an active TIJN account is considered as payment to a New User. The recipient can accept the money within 15 days by signing up to TIJN or adding the additional email address or phone number to an existing TIJN account. After 15 days the payment is cancelled and the funds are returned to the sender.

7. A $299.99 weekly rolling limit is applied for payments. Payments can be cancelled within 10 minutes after they are initiated. Cancelled payments should be recorded in the database.

8. A user's identity is confirmed if her email addresses and phone numbers are verified. If a user's identity has not been confirmed yet, the limit on the funds that he/she can send to their bank account is $999.99 per week. Users with confirmed identities can transfer up to $19,999.99 per week. The single largest amount that can be sent to a bank account at one time is $499.99 for users without confirmed identities and $9,999.99 for users with confirmed identities. Note that users can't transfer funds to bank accounts that they haven't verified.

9. In the TIJN Network, two or more users can be associated with the same bank account in case of joint accounts.

10. In the TIJN Network, two or more users can be associated with the same bank account in case of joint accounts.

11. Users' payment history should be organized in monthly statements. Also, search functionality should be provided (e.g. search for an account, transaction, search for total amount of money sent/received by date, by month, by recipient etc.).

You can make further assumptions but: (a) they should not be in contradiction with the assumptions described above, and (b) they have to be clearly stated in your report.

Note that the above description contains both database requirements and functional requirements. In designing your ER schema, you need to make sure that you have added the constructs that are needed to store the data required by the application. The functional requirements will be implemented at the application level.