

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. 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 WALLET Payment Network

The following specifications are intended as a guide. 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 if they are necessary. 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

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

1. Users can sign up with WALLET (that is, create an account with WALLET) by providing their name, SSN, an email address, and a phone number. Only one phone can be recorded for each WALLET account, but a WALLET account can be associated with multiple email addresses. Email addresses and phone numbers can be recorded in the database but they should be verified before they can be used – i.e., a user should prove that she owns an email address or a phone number by entering correctly the code that is sent via email or SMS respectively. A user's identity is confirmed if her email addresses and phone number are verified.
2. Users can link multiple bank accounts with their account but they should define one of them as the primary funding source. Bank accounts are identified by the bank ID and the account number. Bank accounts have to be verified before they can be used to transfer money to and from the bank accounts to the user's WALLET account. Users

can verify a bank account by reporting the very small amount that WALLET deposited to that account. Users are also able to unlink a bank account.

3. In the WALLET Network, two or more users can be associated with the same bank account in case of joint accounts.
4. There are two forms of actions that WALLET supports:
 - a. Send money to a user
 - b. Request money from users
5. **Send money:** This type of payment can be done by specifying the recipient's email address or phone number, the amount, and a memo (optional). Every "send money" transaction has an Id.
6. **Request money:** A user can request money from others by specifying their email addresses or phone number, 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. Every "request money" transaction has an Id.
7. A payment to an email address or phone number that isn't associated with a WALLET account is considered as payment to a new user. The recipient can accept the money within 15 days by signing up to WALLET or by adding the additional email address or phone number to an existing WALLET account. After 15 days the payment is cancelled and the funds are returned to the sender. The same happens for money sent to an email address or phone number which is not verified. For every payment, the date and time it was initiated and the date and time it was completed should be recorded in the database.
8. Payments can be cancelled within 10 minutes after they are initiated. Cancelled payments should be recorded in the database as well as the reason they are cancelled.
9. 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 (actions related to the database). In designing your database schemas, 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.