System design for depositing money into a wallet

System design for withdrawing money from a wallet

Simplified digital wallet system

Who are its users

People needing to transfer money locally or across international borders.
Examples: Wise, Grab Wallet
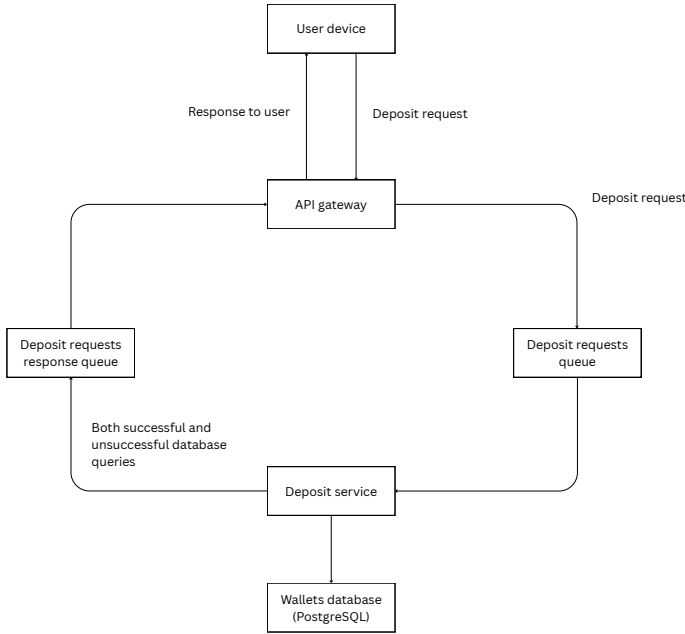
Functional requirements

The user must be able to do the following through RESTful API calls.

1. Deposit money into his wallet.
2. Withdraw money from his wallet.
3. Transfer money from one wallet to another.
4. Retrieve the balance of his wallet.
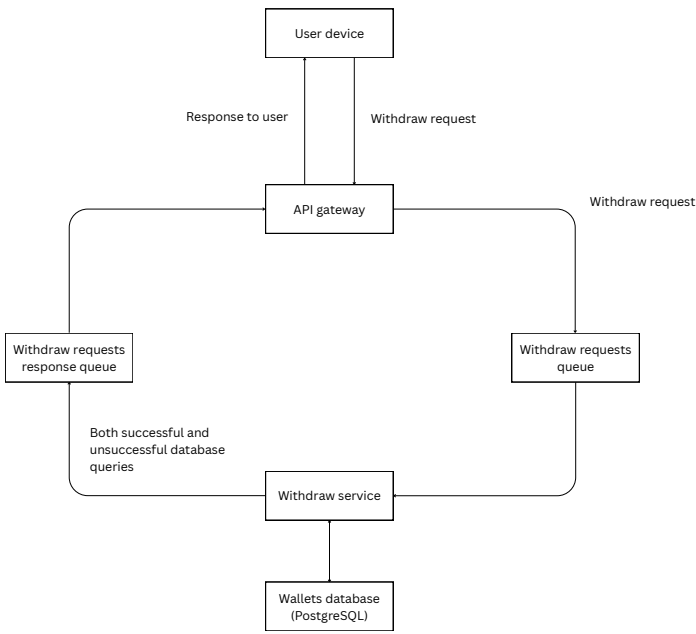5. Retrieve the transaction history of his wallet.

Non-functional requirements

1. System is greatly simplified such that it can be coded up in 3 days with the time of only 1 person for job assessment purposes.
2. System need not be secure.
3. Not try to achieve high availability, scalability and durability as it would be too difficult to build with limited resource.
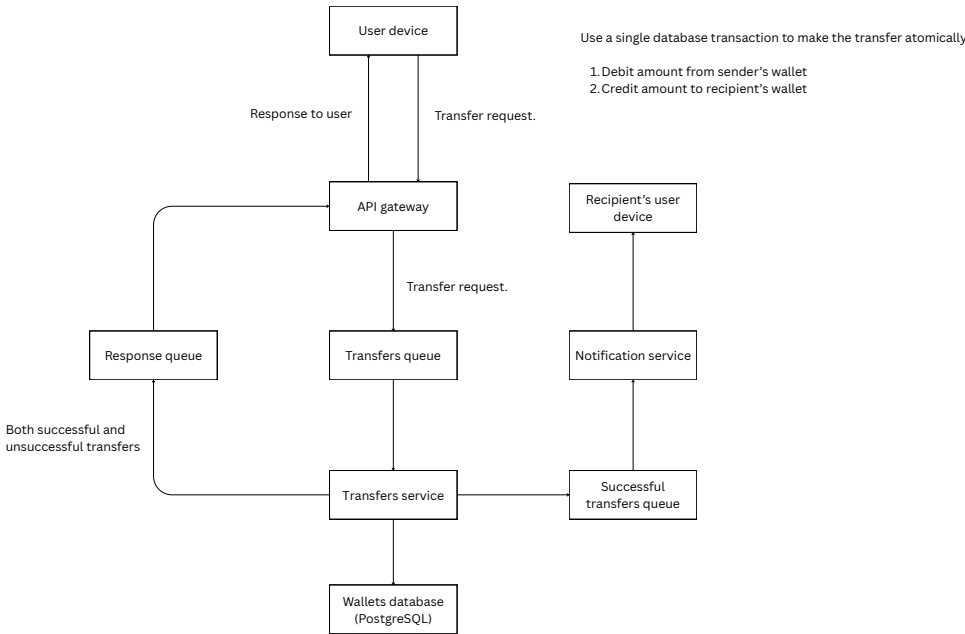
Deposit diagram

User device
— Response to user
— Deposit request

API gateway
— Deposit request

Deposit requests response queue

Deposit requests queue

Both successful and unsuccessful database queries

Deposit service

Wallets database (PostgreSQL)

Withdraw diagram

User device
— Response to user
— Withdraw request

API gateway
— Withdraw request

Withdraw requests response queue

Withdraw requests queue

Both successful and unsuccessful database queries

Withdraw service

Wallets database (PostgreSQL)

System design for transferring money from one wallet to another

Use a single database transaction to make the transfer atomically

1. Debit amount from sender's wallet
2. Credit amount to recipient's wallet

User device
— Response to user
— Transfer request.

API gateway
— Transfer request.

Recipient's user device

Response queue

Transfers queue

Notification service

Both successful and unsuccessful transfers

Transfers service

Successful transfers queue

Wallets database (PostgreSQL)

System design for retrieving the balance of a wallet

User device
— Response to user
— Balance request

API gateway
— Balance request.

Balance requests response queue

Balance requests queue

Both successful and unsuccessful database queries

Balance query service

Wallets database (PostgreSQL)
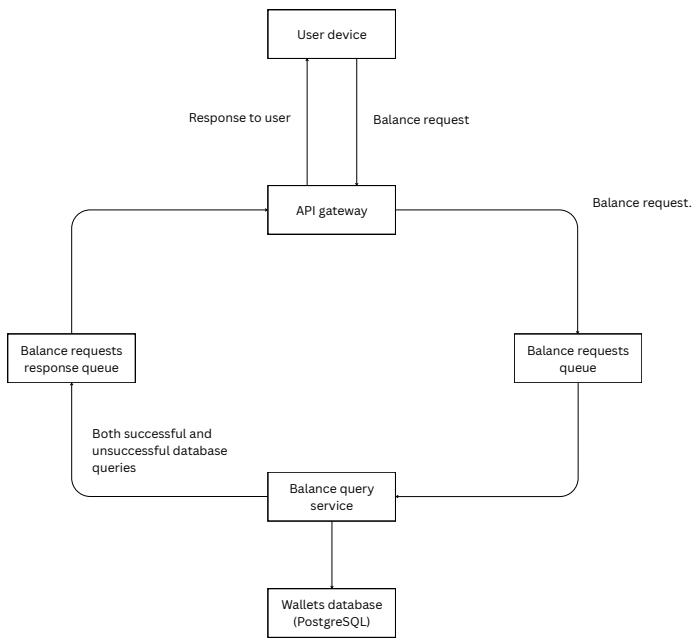
System design for retrieving the transaction history of a wallet

User device
— Response to user
— Transaction history request

API gateway
— Transaction history request

Transaction history response queue

Transaction history requests queue

Both successful and unsuccessful database queries

Transaction history query service

Wallets database (PostgreSQL)