

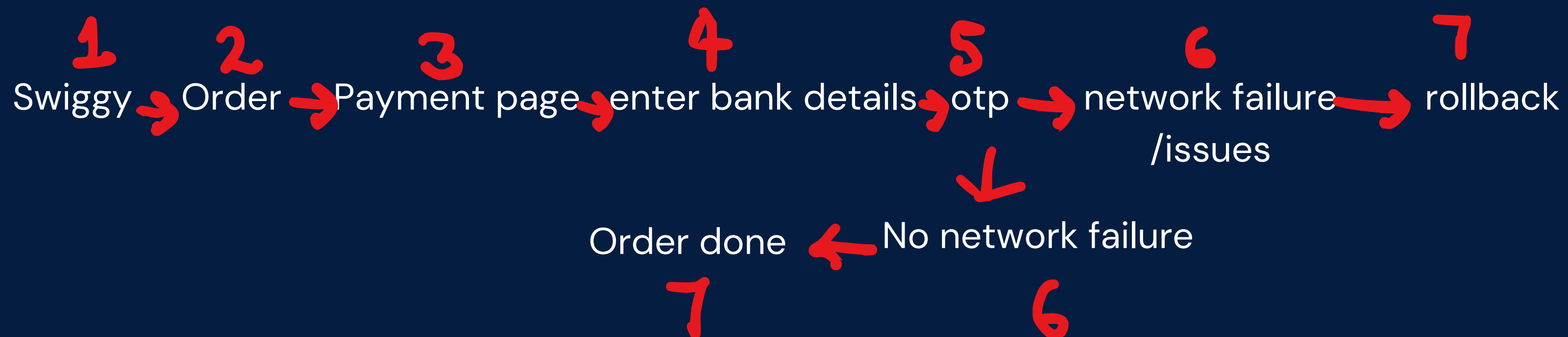
LET'S START WITH DBMS :).

Transaction And Concurrency control

Transaction is a logical unit of work that comprises one or more database operations (like Read/write/commit/rollback). In a transaction both read and write operations are fundamental actions that ensure ACID properties of transactions (data consistency and integrity)

Read(R) → A read operation involves retrieving/fetching data from the database.

Write(W) → A write operation in a transaction involves modifying data in the database



LET'S START WITH DBMS :).

Consider an example of a banking application where a customer Ram wants to transfer 100rs to Shyam

Step 1: Begin Transaction

Step 2: The application will fetch/read the current bal of the Ram → **R(Ram)**

Step 3: The application will calculate the new balances after the transfer →

Ram=Ram-100

Step 4: Update it in the temporary Storage/transaction logs → **W(Ram)**

Step 5: The application will fetch/read the current bal of the Shyam → **R(Shyam)**

Step 6: The application will calculate the new balances after the transfer →

Shyam=Shyam+100

Step 7 : Update it in the temporary Storage /transaction logs → **W(Shyam)**

Step 8: If all updates are successful and there are no errors, commit the transaction to make the changes permanent. Logs are maintain till the occurence of commit, after that log files are deleted and database is updated till step7 → **COMMIT**

Step 9: If an error occurs during any step rollback the transaction to revert changes made within the transaction. → **ROLLBACK**

LET'S START WITH DBMS :).

Transaction states

Case-1

Case-2

BEGIN

R(Ram)

Ram=Ram-100

W(Ram)

R(Shyam)

Shyam=Shyam+100

W(Shyam)

COMMIT

ROLLBACK

ACTIVE

ACTIVE

PARTIALLY COMMITTED

SUCCESSFUL

COMPLETED

FAILED

CANCELLED/ROLLBACK

COMPLETED

LET'S START WITH DBMS :).

Transaction And Concurrency control

Concurrency control ensures that multiple transactions can run concurrently without compromising data consistency.

Example : Consider a banking system where two transactions are happening concurrently

1. Ram giving Shyam 100rs
2. Shyam giving Ram 50rs , the data should be consistent for both transactions

Some techniques used here are:

- Locking
- Two-Phase Locking (2PL)
- Timestamp Ordering