LLD Interview Problem – Splitwise Design (Swiggy SDE-1)

Problem Statement

Design a simplified version of a Splitwise-like expense sharing system. The goal is to track shared expenses between users and determine who owes whom. This problem focuses on Low-Level Design (LLD) using Java and object-oriented principles.

Scope & Assumptions
- No UI or database implementation is required.
- Focus only on in-memory data structures.
- No concurrency handling required.
- Only equal expense splitting is supported.
- All users are global (no groups).

Feature 1: User Management
Each user has a unique ID and a name. Users are pre-created and available in the system.

Example:
User(1, "Alice")
User(2, "Bob")
User(3, "Charlie")

Feature 2: Add Expense
A user can add an expense specifying:
- Total amount
- Paid by user
- List of users involved in the expense

Only equal splitting is supported.

Example:
Alice pays 300 for Alice, Bob, Charlie → each owes 100.

Feature 3: Balance Tracking
The system should track net balances between users. If Bob owes Alice ■100, it should be stored internally as Bob → Alice : 100.

Feature 4: Show Balances
Display all non-zero balances in a readable format. If there are no balances, display:
“No balances”

Feature 5: Simplify Debts
The system should minimize the number of transactions by simplifying debts.

Example:
Bob owes Alice 100
Charlie owes Bob 100

After simplification:
Charlie owes Alice 100

Expected Design (LLD Focus)

Candidates are expected to design the following core components:

Classes:
- User
- Expense
- Split
- ExpenseManager

Key Methods:
- addExpense()
- showBalances()
- simplifyBalances()

Sample Input
Users: Alice, Bob, Charlie
Expenses:
Alice pays 300 for Alice, Bob, Charlie
Bob pays 150 for Bob, Charlie

Expected Output
Charlie owes Alice 50
Charlie owes Bob 25

After simplification:
Charlie owes Alice 75

Evaluation Criteria
- Correct object-oriented design
- Proper use of Java collections
- Clean separation of responsibilities
- Readable and extensible code