

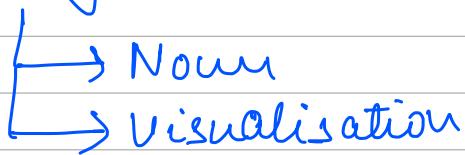
## Splitwise

- 1) Class diagram
- 2) Schema design

Start @ 9.05

5 mins to go over requirements

### Class diagram



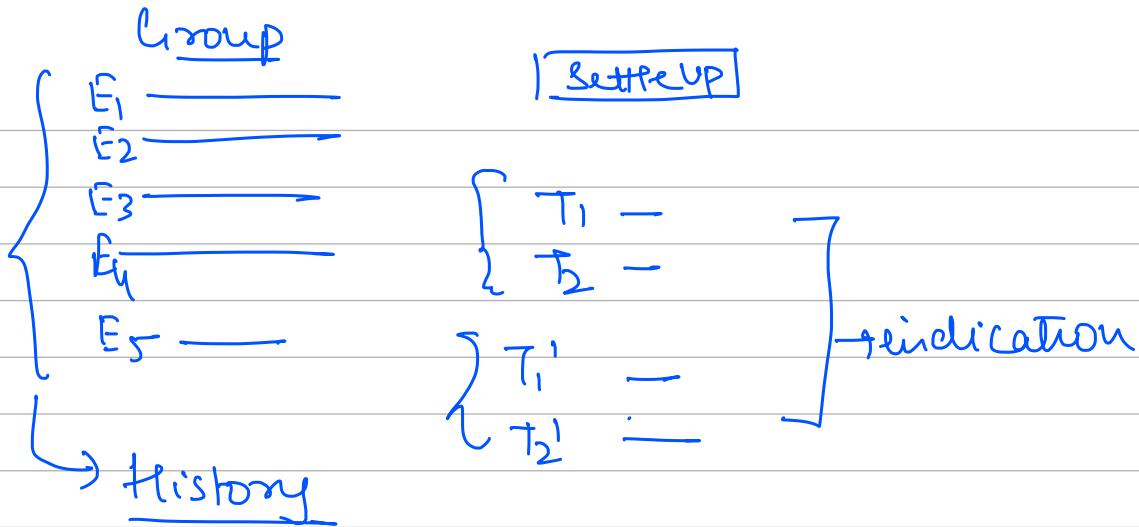
⇒ Go through all the requirements & in each requirement find out the Nouns

→ Check if we need to store the data for this noun in the table or not

### Nouns

- 1) User → profile
- 2) Expense
- 3) Group
- 4) Transaction

→ Do we need to store the data for transaction ⇒ NO



eg **Expense 1** A paid 2000  
A B C D

Had to pay  $A = 500$

$$B = \underline{500}$$

$$C = \underline{500}$$

$$D = \underline{500}$$

extra paid

$$A : 2000 - 500 = \underline{+1500}$$

$$B : 0 - 500 = \underline{-500}$$

$$C : 0 - 500 = \underline{-500}$$

$$D : 0 - 500 = \underline{-500}$$

C cleared Settle UP

$$\sum T_i \quad C \rightarrow A \quad 500$$

↓      ↓  
Paid less    Paid extra



$$\text{Paid By } C = \underline{500}$$

$$\text{Had to pay } A = \underline{500}$$

$$\underline{\underline{C = 0}}$$

Reverse Expense

$$\underline{\underline{C \rightarrow A}}$$

Paid extra

$$C : 0 - 500 + 500 - 0 = 0$$

Paid extra

$$C = 500 - 0 = \underline{+500}$$

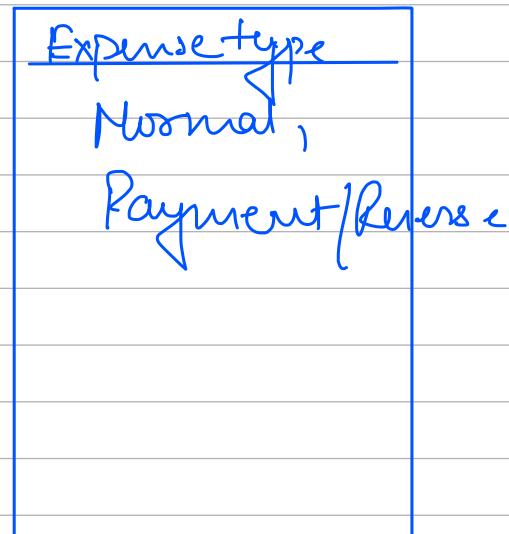
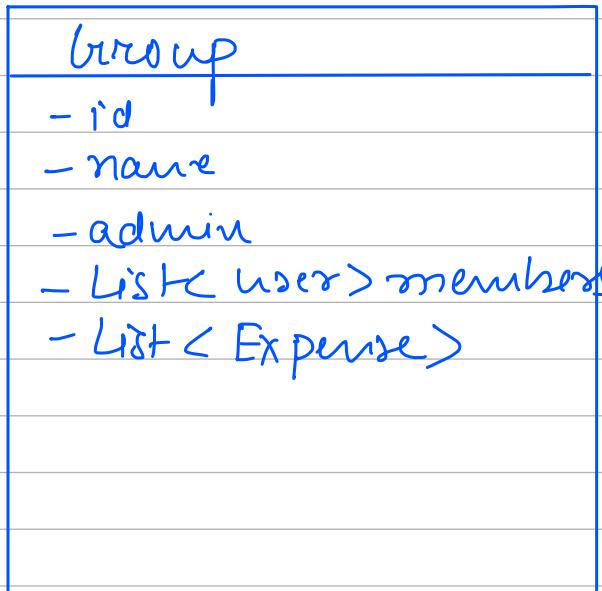
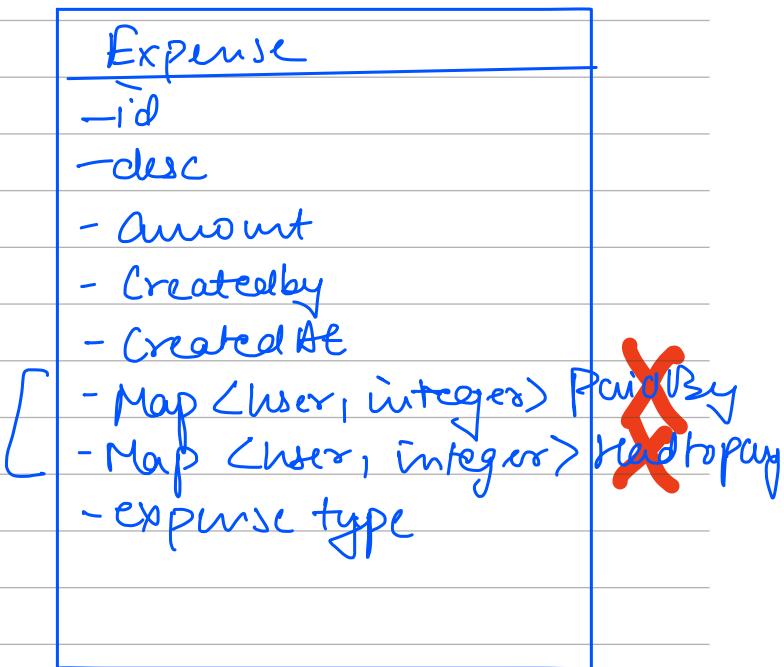
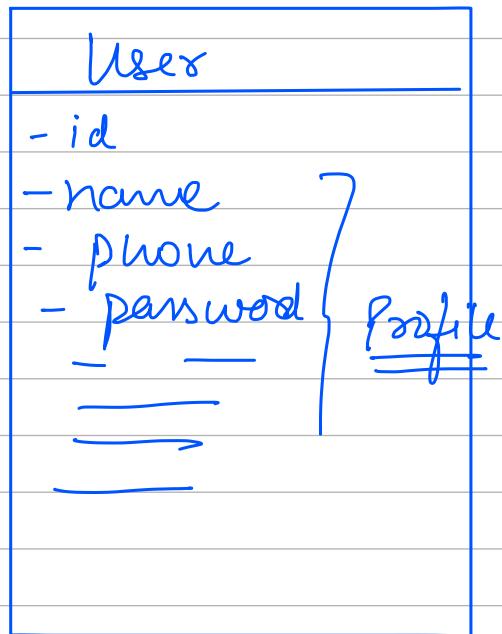
$$A = 0 - 500 = \underline{-500}$$

Paid extra

$$C = 0$$

$$A = \underline{+1000}$$

# We will create a dummy expense which basically reverse the previous expense to make overall paidBy and hadToPay amount for involved users to be zero

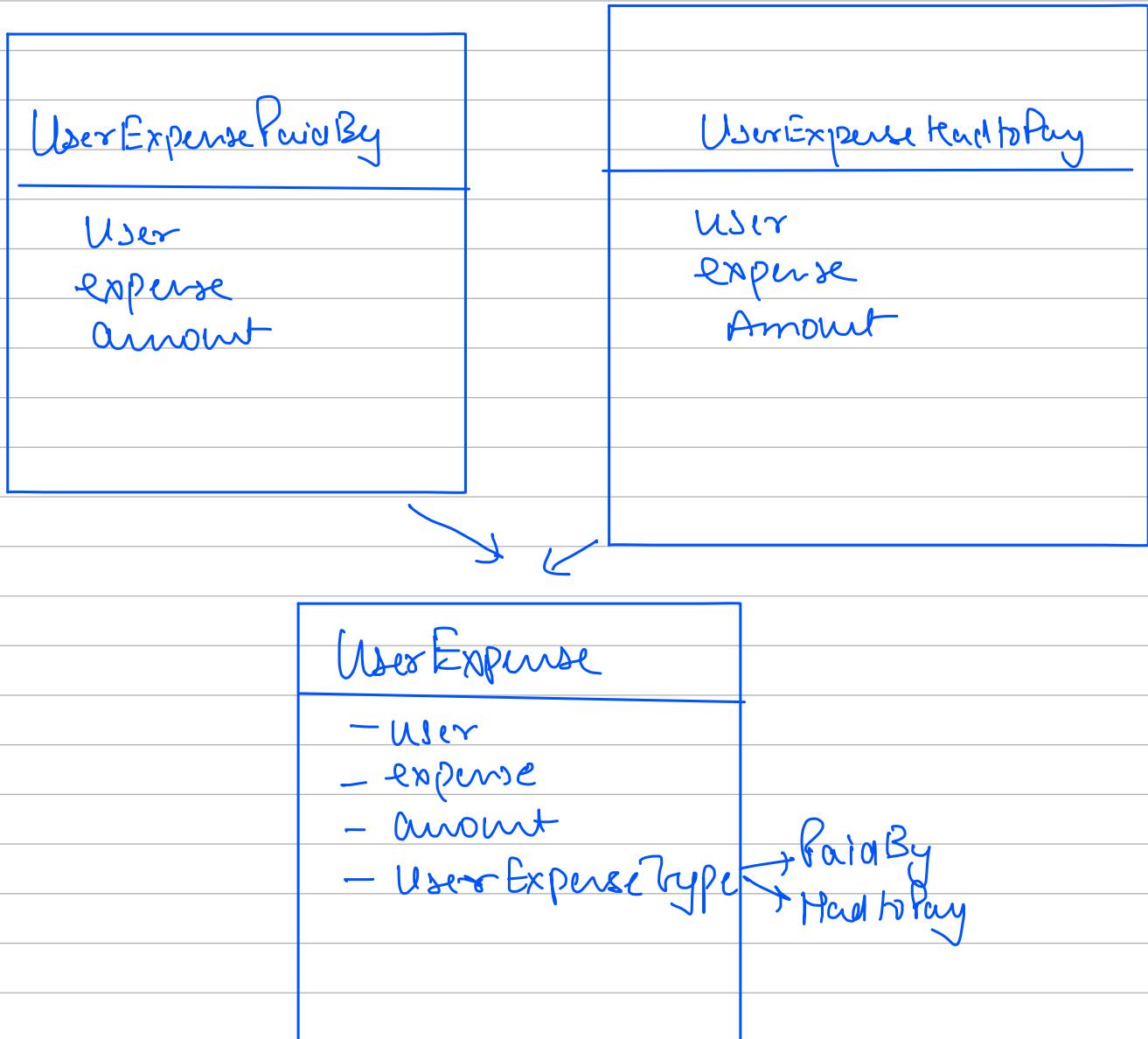


# How to store relation b/w A and B

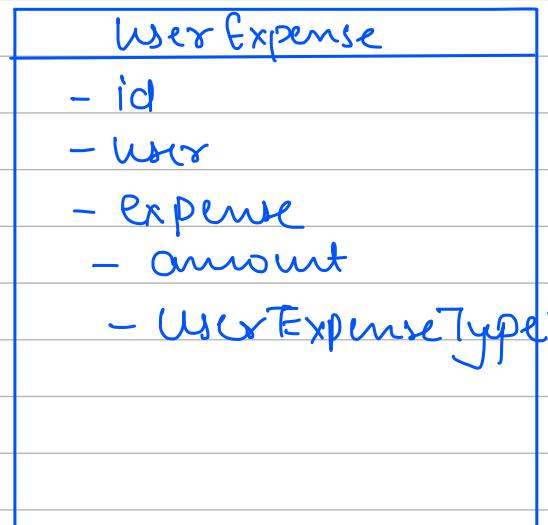
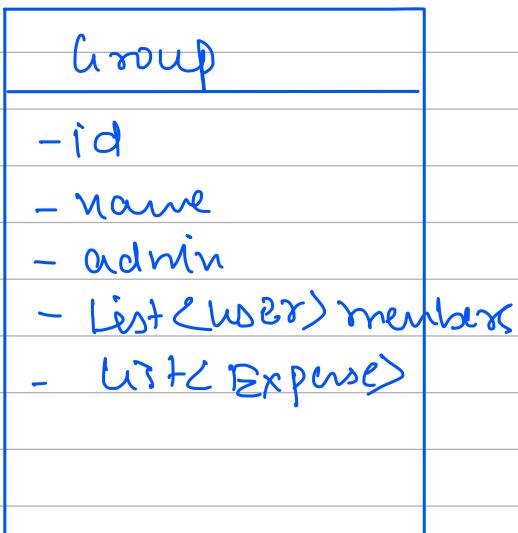
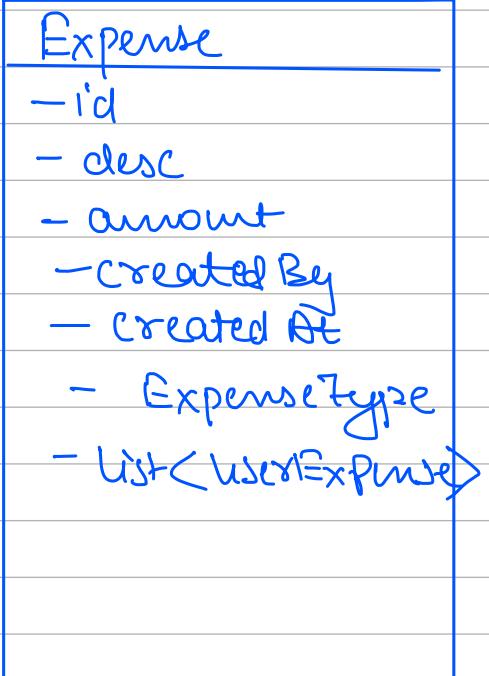
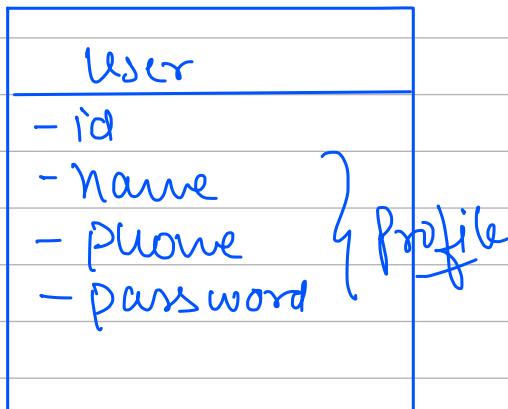
A

B

- 1) A is object in B
- 2) B is object in A
- 3) separate class



## Class Diagram



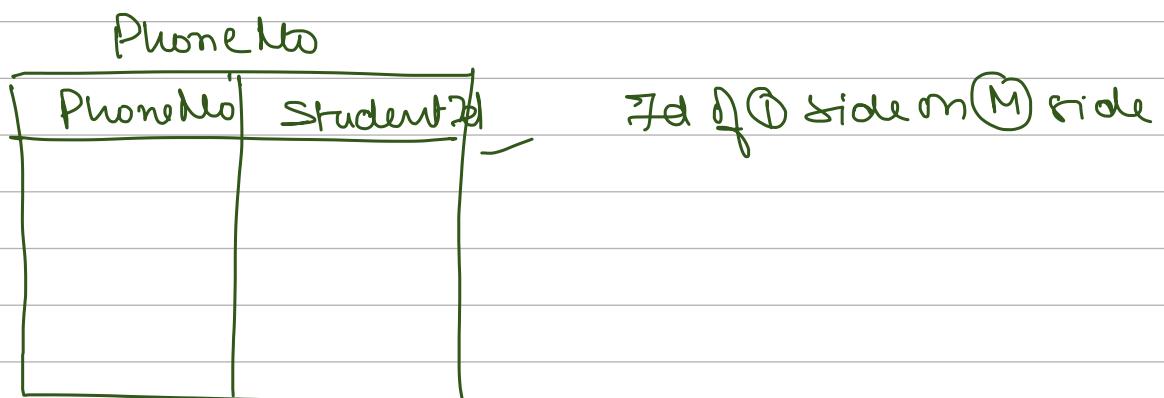
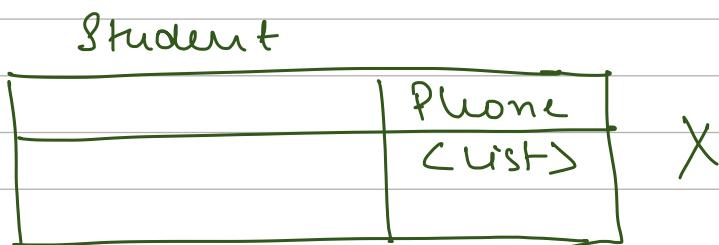
## # Schema design

- ⇒ Create the tables for all the classes in the Class Diagram
- ⇒ Create columns for primitive types directly
- ⇒ For non primitive types find the cardinality & apply rules

1:1 : 1 to 1 on one side on the other side

1:N : 1 to N on one side on the other side

M:M: Mapping table



Splitwise

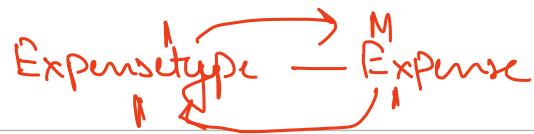
#

Users

id	name	Phone	Password
----	------	-------	----------

Created By — Expense 1:N

## # Expense



id	dese	amount	created-at	created-by-id	ExpenseType Id
					Expense $\xrightarrow{\text{M}} \text{Group}$ GroupId

Note: If lot of expenses have group-id as NULL then it will be sparse table

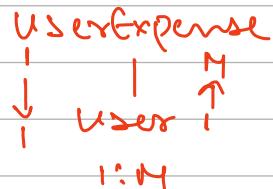
# Group  
z) If this is a sparse table, go with Mapping table

id	name

## # UserExpense



id	amount	expense-id	user-id



## # UserExpenseType

id	value

## # ExpenseType

id	value

## Group Expense

GroupId	ExpenseId

## Assignment

Complete rest of the schema design