Machine Coding : Splitwise

(Will Start at 9:10PM)

Sar Query

- 20 min → Discussion to Design
- 1hr 30 min
- 20 min : Code Review + Follow Ups + Questions

① Take a lock

② Change the ...

③ Release

How will you handle ...

How to approach MC

① Code all Models

② Go req by req

This Week

(Class Diag / Schema Design)

Splitwise-1 : Design, Requirements

-2 : Implement CLI Input Parser (Command DP)

-3 : Implementing the logic to settle up exp

-4 : AM (DSA Problem)

① Design a Distributed Cache (

② Design an Email Campaign Mgmt System (Mailchimp)

③ Concurrency - 4

Revision Class

① Design Patterns

② Concurrency

Requests

① 1 week break

② Coding Contests

Test Case : Project Building

Leo 19-24

M - 25

P - 26

Design Splitwise

① Revise everything that has been covered

② Hidden DSA Question

③ Kind of Not everyone knows

④ ...

(Command Executor App)

⑤ Spring Boot EXE

(Interacting with DB using Spring Boot)

Design Splitwise

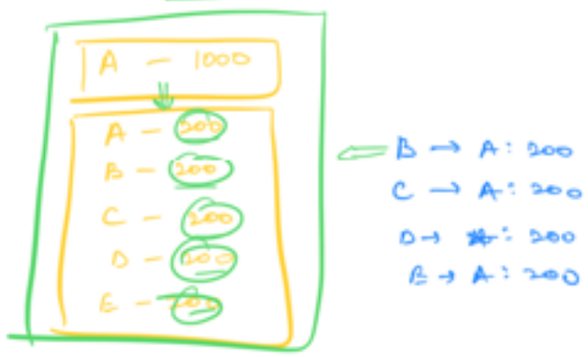
① Overview

↳ Tell what you know

OR ⇒ Ask for a high level understanding

Expense Sharing App

Go for a Trip (A B C D E)



B → A : 200
C → A : 200
D → A : 200
E → A : 200

Create Groups (Goa)

Goa Trip → A, B, C, D,

Gathering Reqs

⇒ Users should be able to register to app?

⇒ Every user has a name, phone No, password

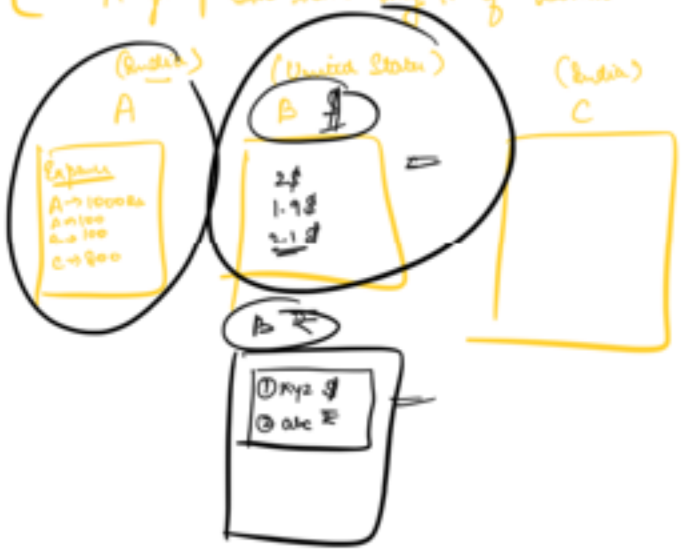
⇒ Expenses can either be Created within a group or amongst a list of users.

⇒ Every group has a name and a

description

⇒ Only admin can add members to group

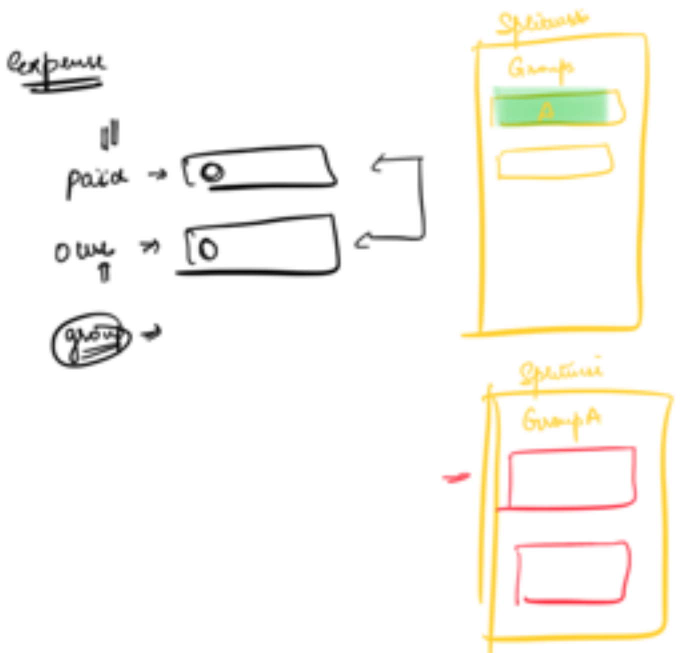
⇒ A group can have any # of admin



⇒ Support for currency

⇒ Every expense will have a base currency of the user who created the expense

⇒ Every other users see the expense in their native currency

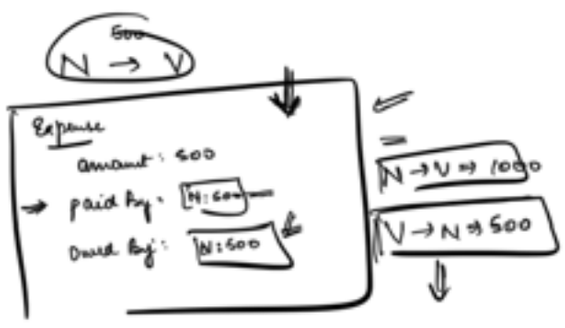


Settle Up

→ Either with a friend →

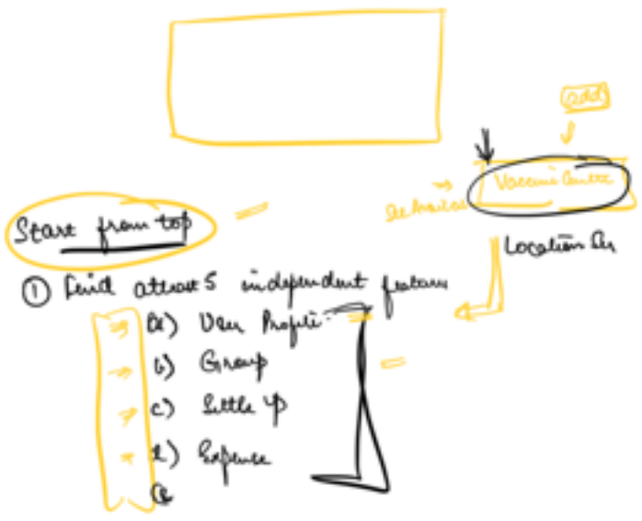
→ Either within a group

Ap person has the amount of money that they need pay / receive as 0 is owed b/w them



N → V ⇒ 1000 =

① Naman paid 500 to Vishal =



② You should be able to end to end via only use cases

A → 1000
B → 200
C → 500
D → 1000

A A