Machine Coding

Splitwise - I

Agenda

- Requirements ⎫
- Design ⎬
- How to create a Spring project ⎫
- Add JPA models to our project ⎬

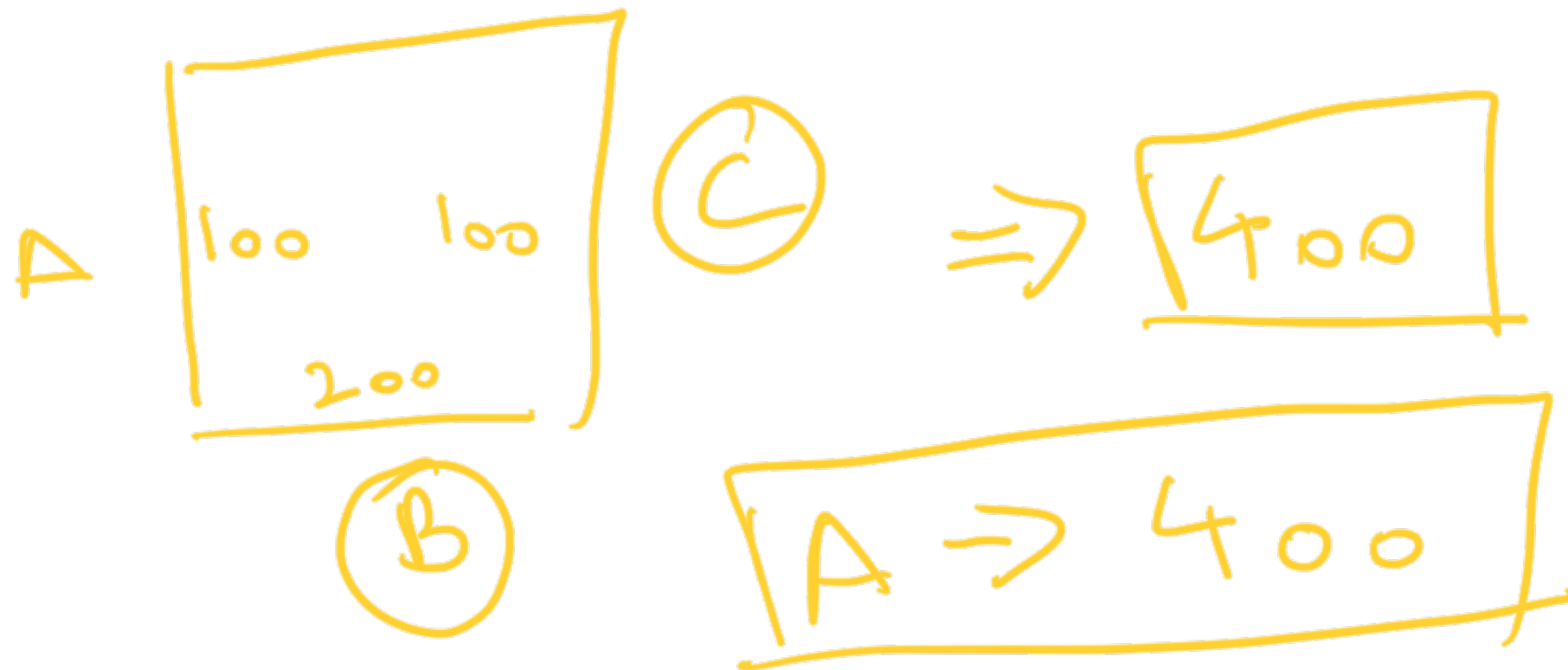- [Postgres]

Starter packs

→ Splitwise
→ Walnut
→ GoDutch

Expense management apps

↓

Sharing

A $\boxed{\begin{array}{cc} 100 & 100 \\ & 200 \end{array}}$ Ⓒ $\Rightarrow$ $\boxed{400}$

Ⓑ $\boxed{A \Rightarrow 400}$

$\left.\begin{array}{l} B \Rightarrow 200 \\ \\ C \Rightarrow 100 \end{array}\right\}$

Splitwise → Add expenses

→ Settle up expenses

→ | Going dutch |

→ Groups

| Scaler
batch 1 | | Batch
2 | → Boxes ne 200

A ones ne lo

A
B
C

→ Only allow people which have no
dves to kcore a group

Data structure

A - B - 100

B - C - 200

D - C - 300

(B)   (C)

200

Settling up

↳ minimize transactions

↳ NP Hard

↳ heuristic

↳ greedy

$\rightarrow$ weighted dig

(A) — India
(B) — UK
} Currency

A — UK $\rightarrow$ | 100 | GBP
A — 100 GBP $\rightarrow$ INR

currency → currency of the person creating the expense

B — India

(B) — | 1000 RS | — | Rupees |

A

→ GBP → | conert to Pounds |
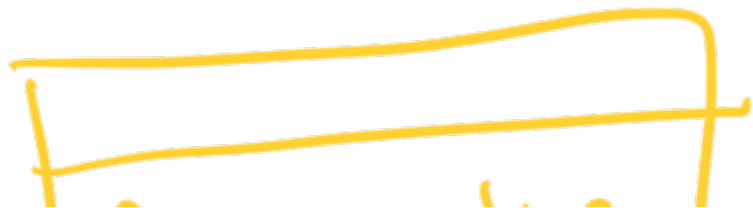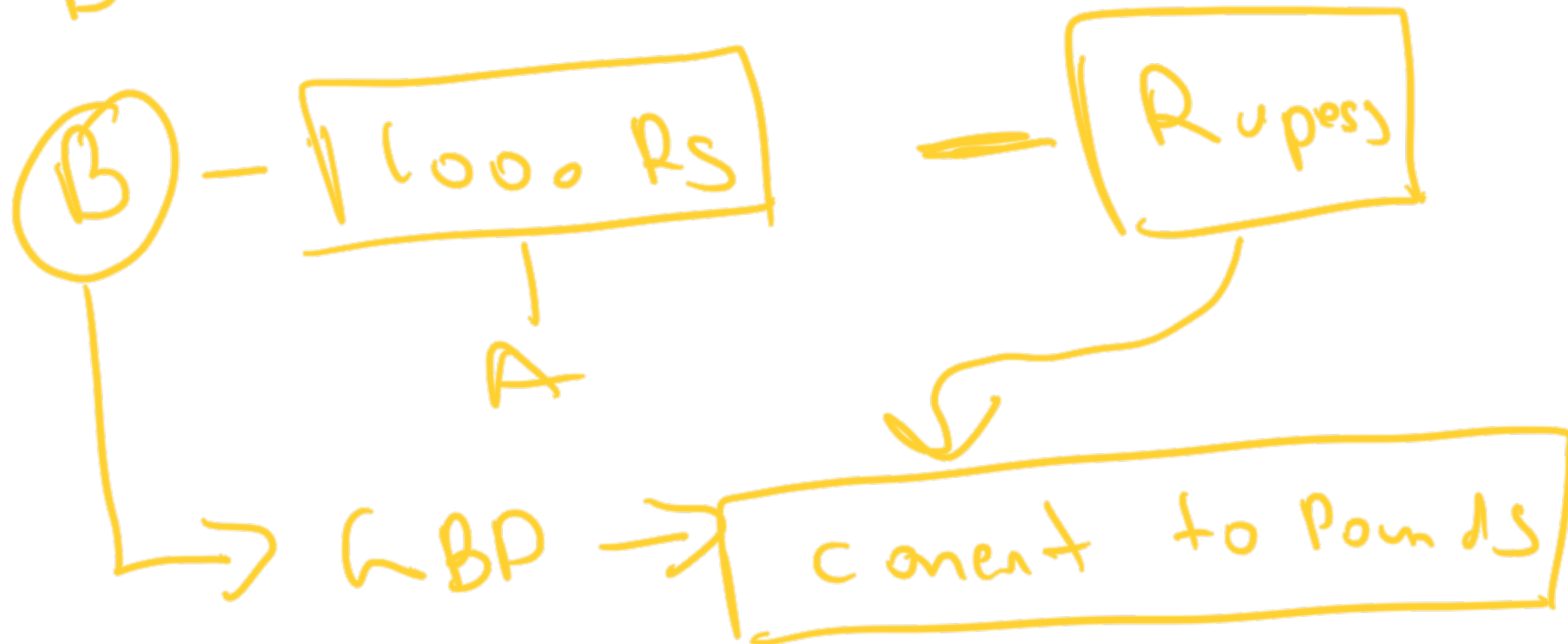
A — 100
B — 100
$\longrightarrow$ * convesion
$\downarrow$

convession rate of
the day

1 GBP — $\dfrac{103\ INR}{75\ INR}$

1 USD — 75 INR

→ Expense

A owes yo

100 INR

- check user currency
  GBP |= INR

  → get conversion rate

  → amount * rate

  return

Entities

A - B = ○

A — B — 100 ○

$$\boxed{A \; paid \; B \; - \; 100} \; \rightarrow \sqrt{A - B} = 0$$

$$\boxed{B \; - \; A \; - \; 100}$$

$$\boxed{Expenses}$$

① $\underline{A \; paid \; 100 \; \cancel{\&} \; B \; oues \; 100}$

② $B \; paid \; A \; 100$

$\Rightarrow$

Class Diagram

User
- name
- email
- phone
- [password]

Group
- name
- users [ ]
- admins [ ]
- expenses [ ]

m:n

m:1
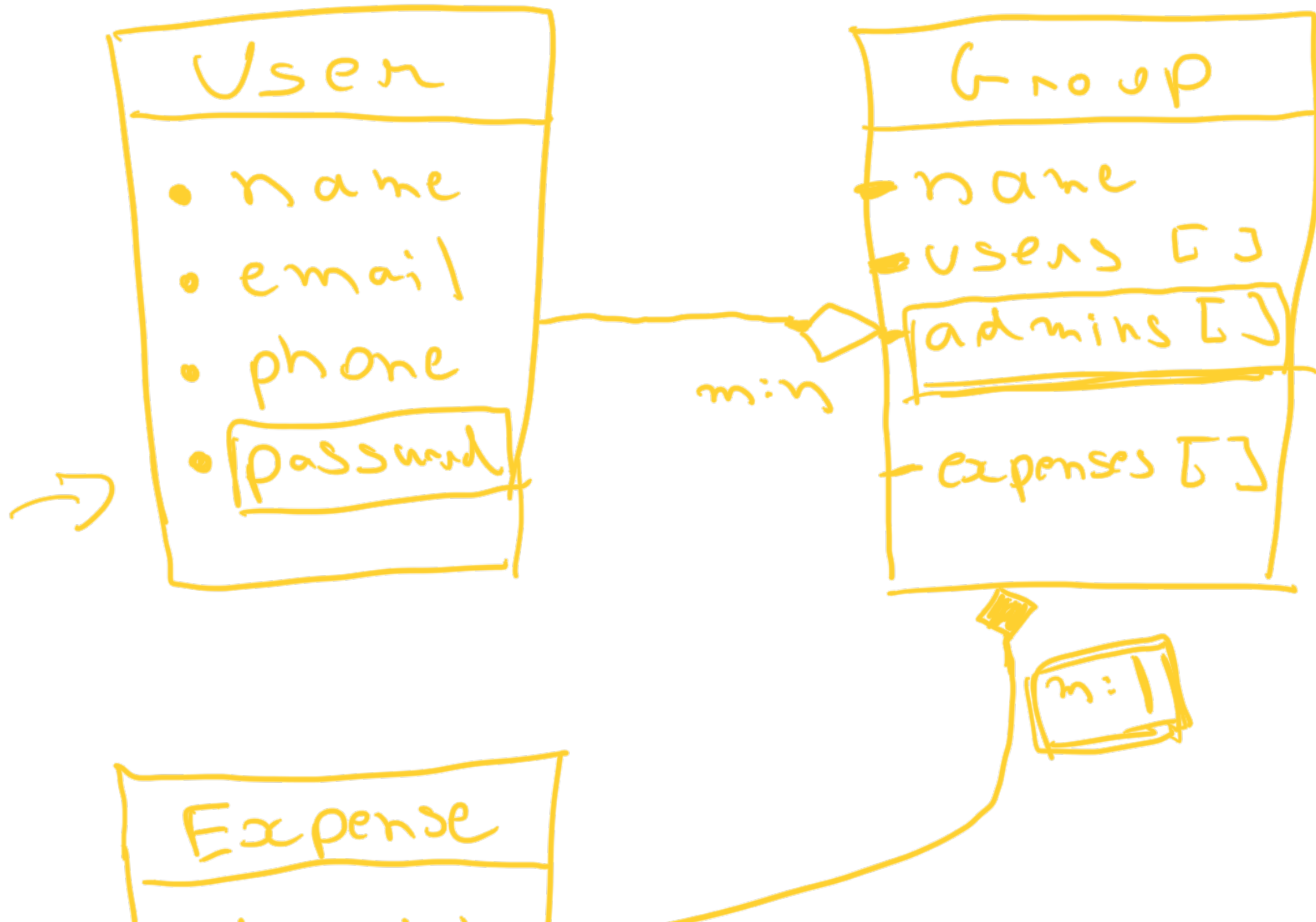
Expense

- description
- amount
- date
- currency
- paid By
- owed By
- created By

JSON

$$100 \rightarrow \boxed{A|B|C|D} \quad 100/4$$

$$\begin{cases} \underline{A = 50} \\ B - 50 \end{cases} \qquad \begin{cases} A \quad - 25 \\ B \quad - 25 \\ C \quad - 25 \end{cases}$$

D   -25

Expense

amt. — 100

Paid By — $\begin{cases} A - 50 ✓✓ \\ B - 50 ✓✓ \end{cases}$

Owed By → $\begin{cases} A - 25 ✓ \\ B - 25 ✓ \\ C - 25 ✓ \\ D - 25 ✓ \end{cases}$

y

C - A - 2S

A - C = 0

⌐→ Expense

5 : 35 - 6 : 00

Group ⇒ admins

If current user in admins.
show option

check if calling user is an admin