Machine coding - Splitwise

① Design Splitwise

- ✓ Overview
- ✓ Requirements
  - - - - - 10 - 15 - - - -
- — Class diagram
- — Schema diagram
- — Command

Implementation

→ Password ✓ → BCrypt

→ Accept in at

accept input from command line ✓

→ DSA problem

Splitwise

→ Expense management Sys.

500	300	200
A	B	C

1000

1000

Expense mgmt.

- Splitwise
- Walnut

Dinner
_ _ _ _ _ _

A → B = 100
C → A = 100   } 2

_ _ _ _ _ _

C = B   } 1

⟹ 100

settle up

A — B — 500
B — A — 500

$$A \xrightarrow{\text{Son}} B$$

$$A \uparrow \text{Son} \quad B \searrow \text{Son}$$

$$C$$

settle

1.
A → C (300)

B → C (300)

2.
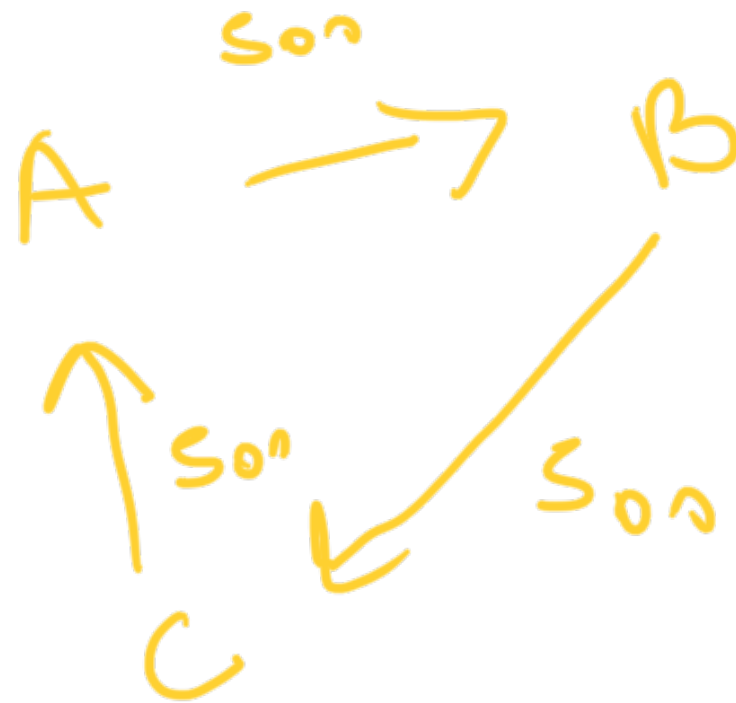
B → C (500)

B —> A (200)

b —> A   100

A — b — 100   A — b —> 0

C — D — 100   b — A — 0

C — O — O

1. Entities
2. Class diagram

Class Break

4:22 — 4:37

10:07

Classes

User

- name
- email
- phone
- avatarURL — [url]
- password

S3

[url]

password → hashed Password
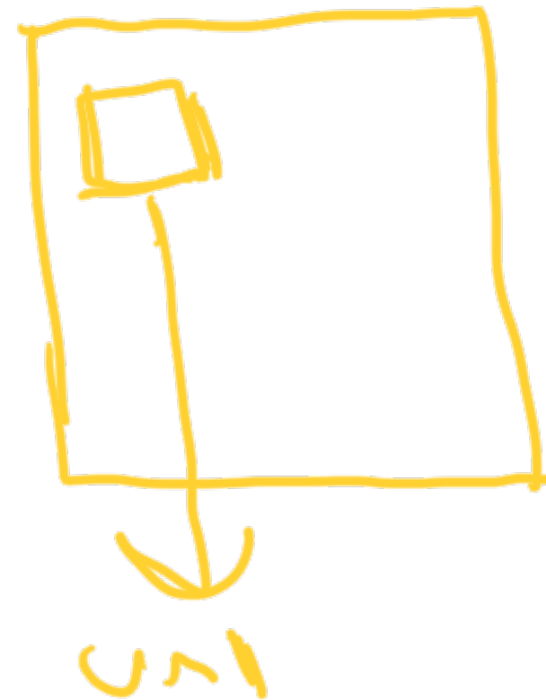
Expense

- description
- amount
- Created By : User [n : 1]
- members
- Group
- currency
- paid By — Map< User, Double >
- Owed By — Map< User, Double >

Group

- name min
- Members [Users]
- Admins [users] min
- Created At
- Created By min

GroupExp

Group
Expense

User

Expense
— User [ ] m
— created By

Group
User [ ] m
User [ ] [ ]
created By

Expense [CB]

User

M

$$M : 1$$

Expense   members $\overset{M}{\text{User}}$

$\underset{M}{}$

$$M : N$$

Expense   Group

$\underset{M}{}$

Group

M
Exponse

1 : M

① 1 : 1 ⟶

M

User <⟶ Email

TemailID   UserId

UserId   Email Id

1

1

2

$\Rightarrow$ 1 : M or M : 1

$\Rightarrow$ M : N $\longrightarrow$ mapping table

USERS

id	name	email	phone	hashed-pw

EXPENSES

id	desc.	amount	currency	created_by_id
1	A	100	$	1
②	B	200	$	2

1000 − 500

expense_id	group_id
1	1

Caveat — NULL values

1:m | m:1 ⟶ Sparse table

⟶ Wastage of space

Mapping table

expense — members

exp_id	user_id
1	1
	2

i	3

Group

id	name	created_At	created_by_id

groups – members

group_id	user_id	is_admin

admins

group

group_id	user_id

expense_ owed_by

expense_id	user_id	amt.

expense_paid_by

expense_id	user_id	amt.

Practice

[
→ Mermaid — Class diagram]
→ Mermaid — Schema diagram

ER ding

→ [Classes] = models *

→ [API design]

→ Try to get rid of Hash maps in

expense