Splitwise

Your objective is to create the backend for an expense sharing application.

An expense sharing application is where you can add your expenses and split it among different people. The app keeps balances between people as in who owes how much to whom.

Examples:

You live with 3 other friends.

You: User1 (id: u1)

Flatmates: User2 (u2), User3 (u3), User4 (u4)

— Example 1:

This month's electricity bill was Rs. 1000.

Now you can just go to the app and add that you paid 1000, select all the 4 people and then select split equally.

Input: u1 paid Rs 1000/- for u1 u2 u3 u4 and needs to be split EQUALLY

For this transaction, everyone owes Rs 250 to User1.

The app should update the balances in each of the profiles accordingly.

User2 owes User1: Rs 250

User3 owes User1: Rs 250

User4 owes User1: Rs 250

— Example 2:

Now, there is a major sale on Flipkart and there is an offer on your card.

You buy a few things for User2 and User3 as they asked you to.

The total amount for each person is different.

Input: u1 paid Rs 1250/- and u2 u3 owe EXACTLY Rs 370/- and Rs 880/-

For this transaction, User2 owes 370 to User1 and User3 owes 880 to User1.

The app should update the balances in each of the profiles accordingly by including the balances from the previous example.

User2 owes User1: 620 (250+370)

User3 owes User1: 1130 (250+880)

User4 owes User1: 250 (250+0)

— Example 3:

Now, you go out with your friends and take your brother/sister along with you.

User4 pays and everyone splits equally. You owe for 2 people.

Input: u4 paid 1200 for u1 u2 u3 u4 and the split PERCENTAGE is 40 20 20 20

For this transaction, User1 owes 480 to User4, User2 owes 240 to User4 and User3 owes 240 to User4.

The app should update the balances in each of the profiles accordingly.

User1 owes User4: 230 (250-480)

User2 owes User1: 620 (620+0)

User2 owes User4: 240 (0+240)

User3 owes User1: 1130 (1130+0)

User3 owes User4: 240 (0+240)

Requirements

User: Each user should have a userId, name, email, mobile number.

Expense: Could either be EQUAL, EXACT or PERCENT

Users can add any amount, select any type of expense and split with any of the available users.

The percent and amount provided could have decimals upto two decimal places.

In case of percent, you need to verify if the total sum of percentage shares is 100 or not.

In case of exact, you need to verify if the total sum of shares is equal to the total amount or not.

The application should have a capability to show expenses for a single user as well as balances for everyone.

When asked to show balances, the application should show balances of a user with all the users where there is a non-zero balance.

The amount should be rounded off to two decimal places. Say if User1 paid 100 and amount is split equally among 3 people. Assign 33.34 to first person and 33.33 to others.

There should be an option to simplify expenses. When simplify expenses is turned on (is true), the balances should get simplified. Ex: ‘User1 owes 250 to User2 and User2 owes 200 to User3’ should simplify to ‘User1 owes 50 to User2 and 200 to User3’.

When a new expense is added, each participant in that expense should get an email telling them that they have been added to an expense, the total amount they owe for that expense. This email should be sent asynchronously (non-blocking) so that the API call doesn't get blocked.

And create a scheduled job that will send an email every week to users. This email should contain the total amount of money they owe to each user.

Each expense can have up to 1000 participants and the maximum amount for an expense can go up to INR 1,00,00,000/-

All API responses should take less than 50 milliseconds.

Optional Requirements

- A way to add an expense name while adding the expense. Can also add notes, images, etc.

- Option to split by share. Ex: ‘User4 pays and everyone splits equally. You pay for 2 people.’ could be added as: `u4 1200 4 u1 u2 u3 u4 SHARE 2 1 1 1`

- A way to show the passbook for a user. The entries should show all the transactions a user was part of. You can print in any format you like.

Things you should do:

- Create a readme.md file explaining the design of your system. This should include the architecture diagram, API contracts and the structure of your classes.
- Create the database schema for this application
- Implement the classes and interfaces required to fulfill the above requirements
- Write the HTTP API endpoints required to fulfill the requirements
- Push your code and readme.md file into a github repository and share the link with us