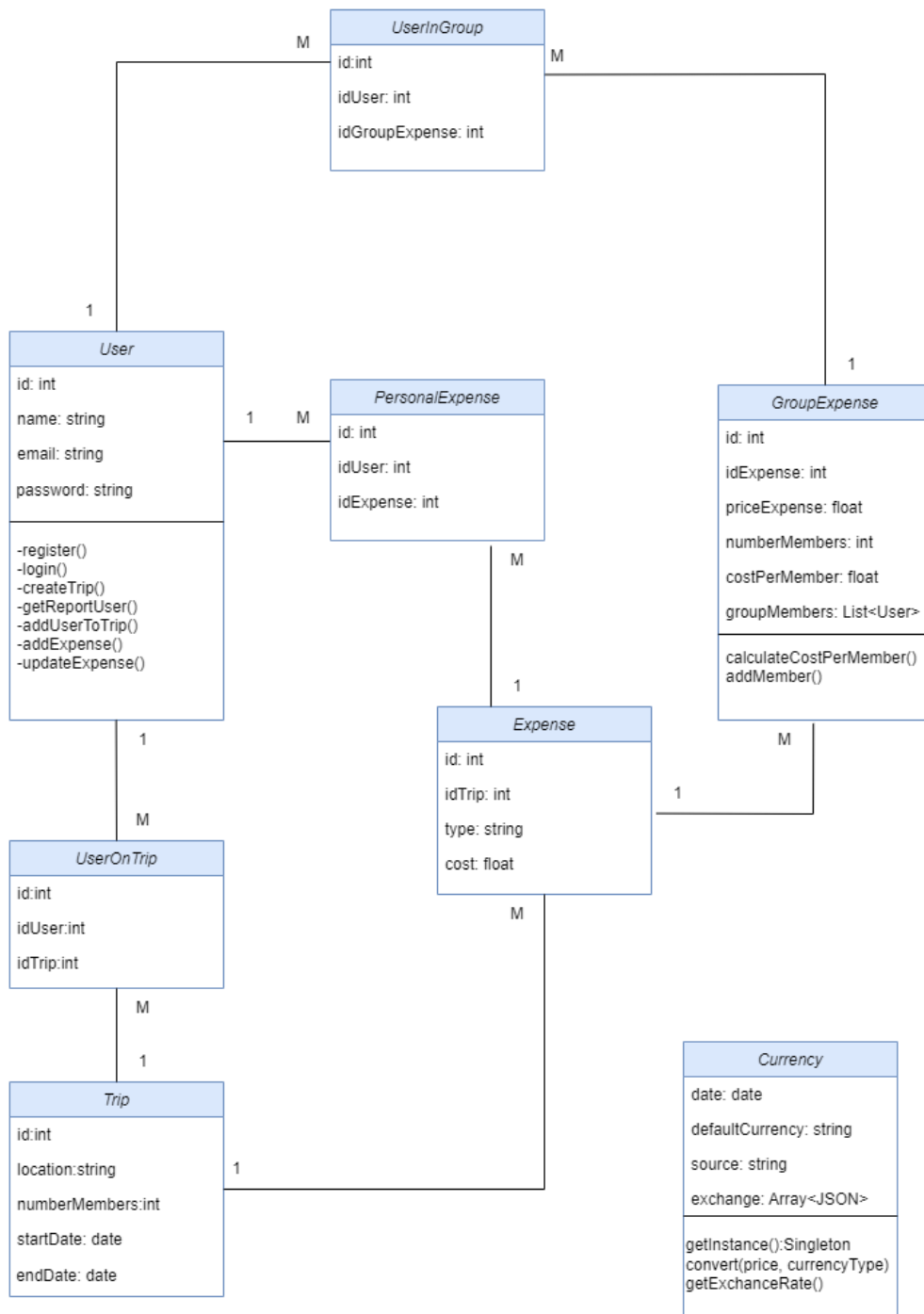


# Group Expense Tracker System

## 1. Class Diagram



## 2.Behavioral Design Pattern – Observer

Notificarea utilizatorilor atunci când face parte dintr-un grup pentru care a fost înregistrată o cheltuială.

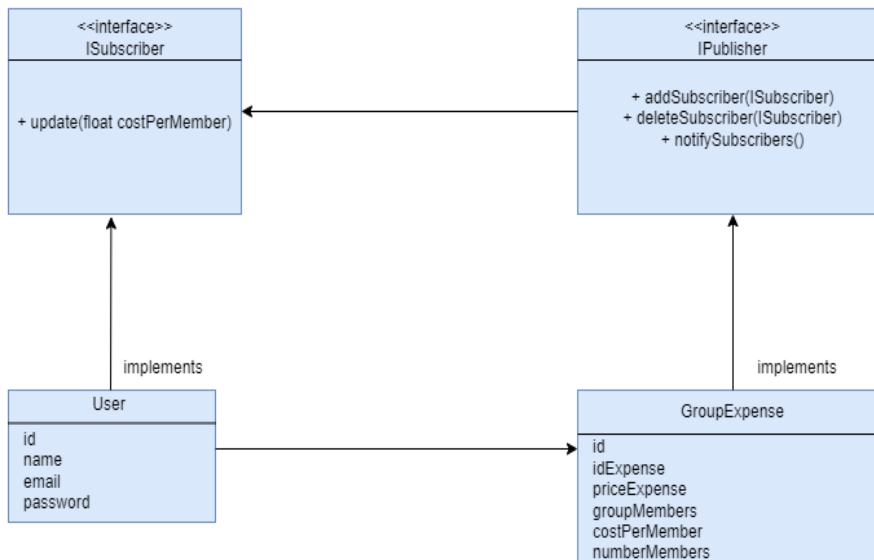
### Problema:

Orice utilizator poate adăuga o cheltuială de grup pentru o excursie. Atunci când o astfel de cheltuială este înregistrată, toți membrii grupului trebuie să fie notificați, primind ca informație suma pe care trebuie să o plătească.

### Soluția:

Se adaugă un Observer Patter cu ajutorul căruia se definește o relație one to many între o cheltuială de grup și membrii grupului. În momentul în care o cheltuială este înregistrată, toți utilizatorii legați de aceasta sunt notificați.

### Diagrama:



## Pseudocod:

```
1  /* Behavioral Design Pattern - Observer
2
3  interface IPublisher is
4      method addSubscriber(ISubscriber)
5      method deleteSubscriber(ISubscriber)
6      method notifySubscribers()
7
8  interface ISubscriber is
9      method update(float costPerMember)
10
11
```

```
12 class GroupExpense implements IPublisher is
13     int id
14     int idExpense
15     float costPerMember
16     int numberMembers
17     float priceExpense
18     List<User> groupMembers
19
20     constructor GroupExpense(id, idExpense) is
21         id = id
22         idExpense = idExpense
23
24     method notifySubscribers() is
25         for subscriber in groupMembers
26             subscriber.update(costPerMember)
27
28     method addSubscriber(ISubscriber) is
29         groupMembers.add(ISubscriber)
30         numberMembers += 1
31         costPerMember = priceExpense/numberMembers
32
33     method deleteSubscriber(ISubscriber) is
34         groupMembers.delete(ISubscriber)
35         numberMembers -=1
36         costPerMember = priceExpense/numberMembers
37
```

```
12 class GroupExpense implements IPublisher is
13     int id
14     int idExpense
15     float costPerMember
16     int numberMembers
17     float priceExpense
18     List<User> groupMembers
19
20     constructor GroupExpense(id, idExpense) is
21         id = id
22         idExpense = idExpense
23
24     method notifySubscribers() is
25         for subscriber in groupMembers
26             subscriber.update(costPerMember)
27
28     method addSubscriber(ISubscriber) is
29         groupMembers.add(ISubscriber)
30         numberMembers += 1
31         costPerMember = priceExpense/numberMembers
32
33     method deleteSubscriber(ISubscriber) is
34         groupMembers.delete(ISubscriber)
35         numberMembers -=1
36         costPerMember = priceExpense/numberMembers
37
```

