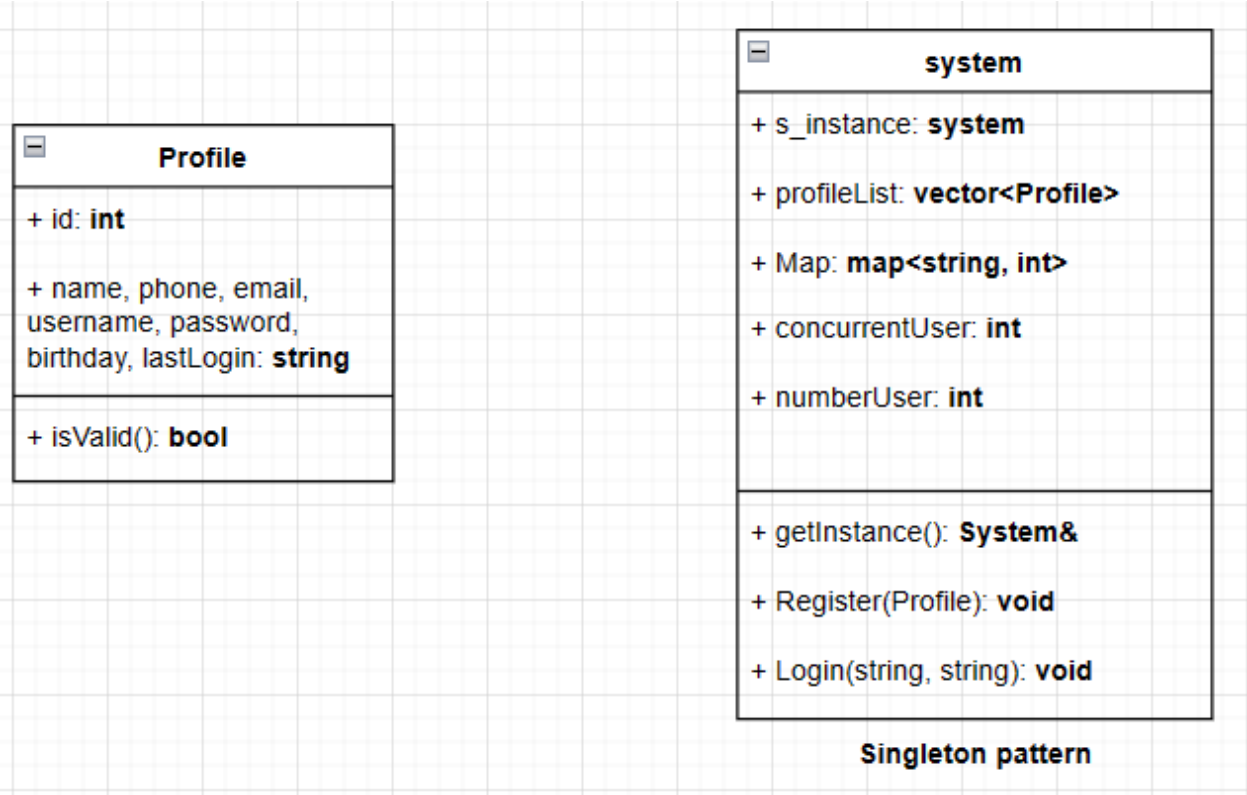
Problem 1: Build login/register system

Class diagram:



In this problem, I'd created 2 classes: "Profile" and "system".

About "Profile" class:

- **id**: profile's id. It is also the index of profileList vector in "system"class.
  - If id = 5 -> this profile has stored in position 5 of profileList.

- **isValid()** : Check for the Profile valid or not.
  - Check username, email, password are null or not.
  - Check password is null or not.

About "system" class:

- **profileList:** This is a vector that store profile whenever a new valid register occurs.

- **Map:** Data structure which is STL in C++ that store pair of (key, value). This data structure is built **based on Red-Black Tree** (or Binary Search Tree) and have **complexity of O(log2(size))**.
    - Whenever a new registers successfully, this Map will **store pair of (username, id), (email, key), (phone, key)**.
        - Ex: If the new register has id = 5, username = "Cake" -> push ("Cake", 5) to Map.

- **concurrentUser:** number of concurrent users in the system. By using this variable, I can control the limit user can login to the system. I choose 10^6 to be the system limit.

- **numberUser:** number of users in the system. I also use this to give new register an id (then increase it later).
    - numberUser = 7 and a new register come -> The system create new profile has id = 7, then numberUser increase to 8.

- **Register(Profile):** This function will process an input register.
    - Check null using "**isValid()**" method in class "Profile".
    - Checked for existing information before. (Ex: input exist username)
    - If register successfully, add new profile to vector and push pairs of (username, id), (email, id), (phone, id) to Map (which cost logarithm complexity).

- **Login(string, string):** This function with 2 parameters as input account and password.
    - Check for key = account has been existing in Map before.

    - If there is a valid key = account in Map, get the value = id was saved before in Map. By using this id, we can access the profile in profileList. Then just compare input password with the profile we just found.
        - **This process have complexity of O(log2(size)) because of finding the key value in Map.**

    - Check for number of concurrent in the system. If reach the limit so this user can not access to the system.

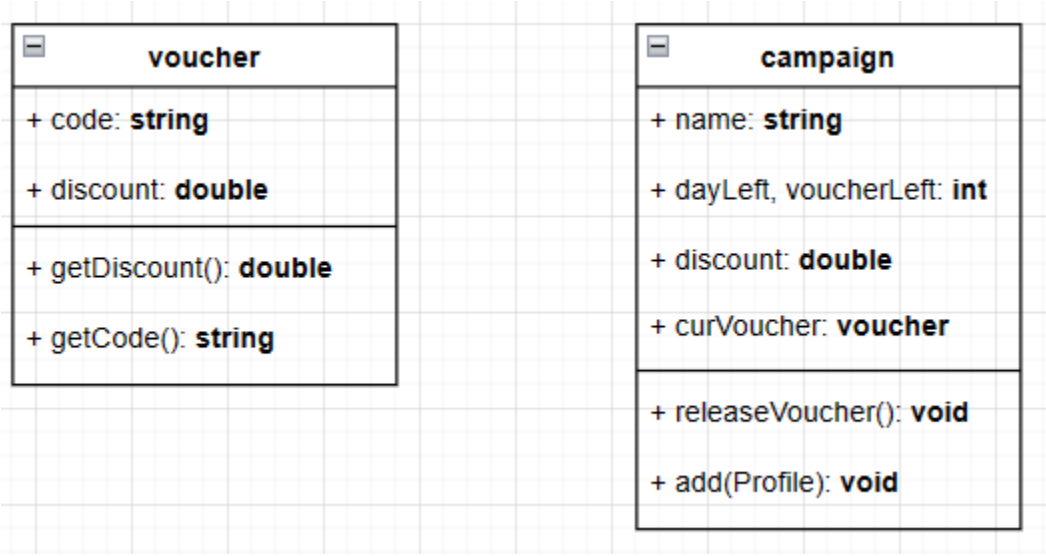
Run the program:

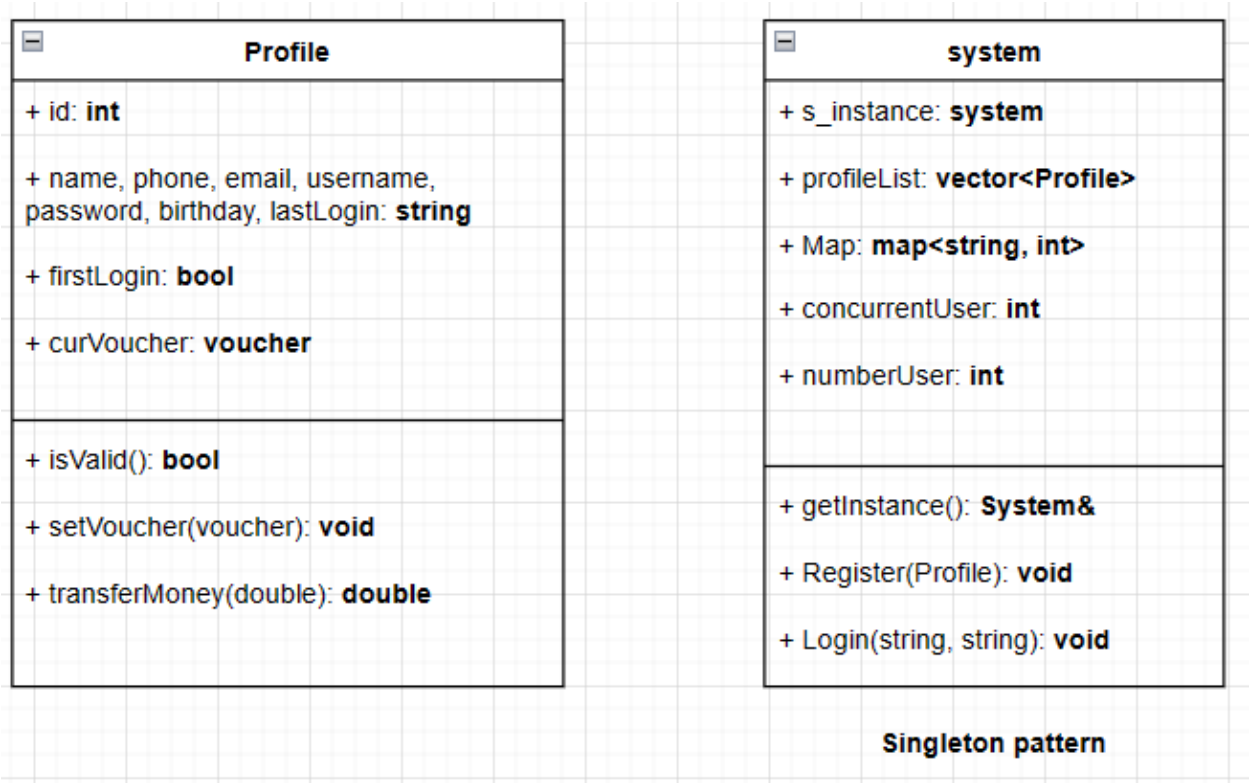
```
WELCOME TO THE SYSTEM!
Input 0/1/2 to exit/login/register the system: 2

Input your full name (or skip by enter):
Input your phone (or skip by enter):
Input your email (or skip by enter):
Input your username (or skip by enter): cake
Input your password (or skip by enter): sss
Input your birthday (or skip by enter):

Register successfully!
Your information has been saved to the system.
--------------------------------------------------------------------------------
WELCOME TO THE SYSTEM!
Input 0/1/2 to exit/login/register the system: 1

Input your account: cake
Input your password: sss

Login successfully!
--------------------------------------------------------------------------------
```

Register then login successfully.

Problem 2: Build promotion system

Class diagram:





In this problem, I'd create more 2 classes "voucher" and "campaign" and keep 2 classes "Profile" and "system".

About "Profile" class:

- I add 2 new methods:
  - **setVoucher(voucher):** When the user valid to give the voucher then receive voucher from the campaign.
  - **transferMoney(int):** Function to calculate the total amount of money need to transfer including using voucher (if any).

About "campaign" class:

- There are basic information like name, discount, number of vouchers. I have made 2 methods:
    - **releaseVoucher**(): This function generate a new voucher then set it to current voucher which is ready to give for next valid login.

    - **add(Profile):** This function give voucher to profile which has been valid.

Run the program:

```
WELCOME TO THE SYSTEM!
Input 0/1/2 to exit/login/register the system: 2

Input your full name (or skip by enter):
Input your phone (or skip by enter):
Input your email (or skip by enter):
Input your username (or skip by enter): cake
Input your password (or skip by enter): sss
Input your birthday (or skip by enter):

Register successfully!
Your information has been saved to the system.
--------------------------------------------------------------------------------
WELCOME TO THE SYSTEM!
Input 0/1/2 to exit/login/register the system: 1

Input your account: cake
Input your password: sss

Login successfully!
Congratulations! You have receive a voucher from Cake with discount of 30%.
Your voucher's CODE: CAKE_DISCOUNT_100
--------------------------------------------------------------------------------
```

New register then login to the system and get the discount.