EXERCISE OVERVIEW

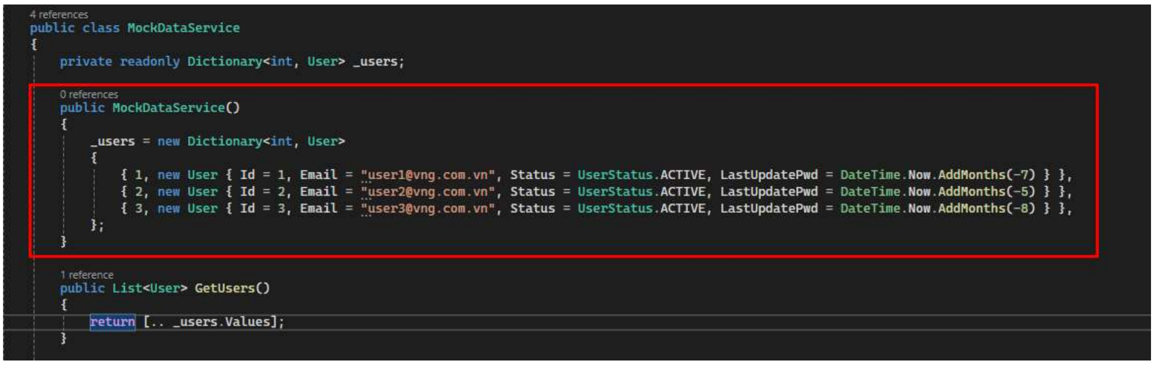
Exercise 1: Develop a Background Job in .NET Core

Background Job use Service Worker to create default settings

Idea: Find users who changed their password last time more than 6 months ago.

(The data assumes there are 3 users, and the email is just for example.)

⇨ Users with emails “user1@vng.com.vn” and “user2@vng.com.vn” need to update their password because it's been over 6 months.

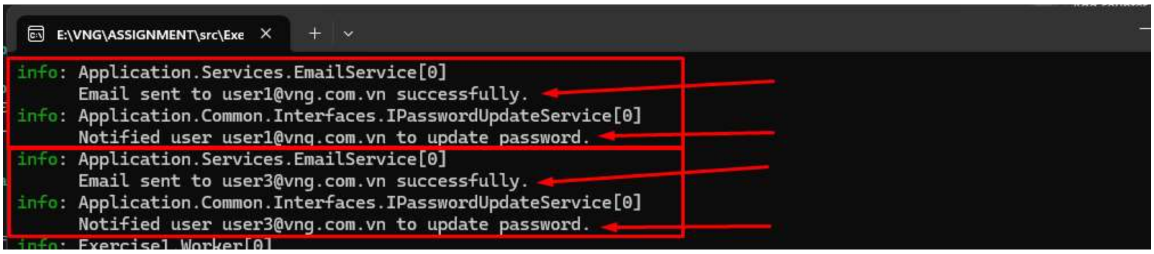


```
4 references
public class MockDataService
{
    private readonly Dictionary<int, User> _users;

    0 references
    public MockDataService()
    {
        _users = new Dictionary<int, User>
        {
            { 1, new User { Id = 1, Email = "user1@vng.com.vn", Status = UserStatus.ACTIVE, LastUpdatePwd = DateTime.Now.AddMonths(-7) } },
            { 2, new User { Id = 2, Email = "user2@vng.com.vn", Status = UserStatus.ACTIVE, LastUpdatePwd = DateTime.Now.AddMonths(-5) } },
            { 3, new User { Id = 3, Email = "user3@vng.com.vn", Status = UserStatus.ACTIVE, LastUpdatePwd = DateTime.Now.AddMonths(-8) } },
        };
    }

    1 reference
    public List<User> GetUsers()
    {
        return [.. _users.Values];
    }
```

Result: 2 log lines show emails sent and user information updated successfully.



```
info: Application.Services.EmailService[0]
      Email sent to user1@vng.com.vn successfully.
info: Application.Common.Interfaces.IPasswordUpdateService[0]
      Notified user user1@vng.com.vn to update password.
info: Application.Services.EmailService[0]
      Email sent to user3@vng.com.vn successfully.
info: Application.Common.Interfaces.IPasswordUpdateService[0]
      Notified user user3@vng.com.vn to update password.
info: Exercise1.Worker[0]
```

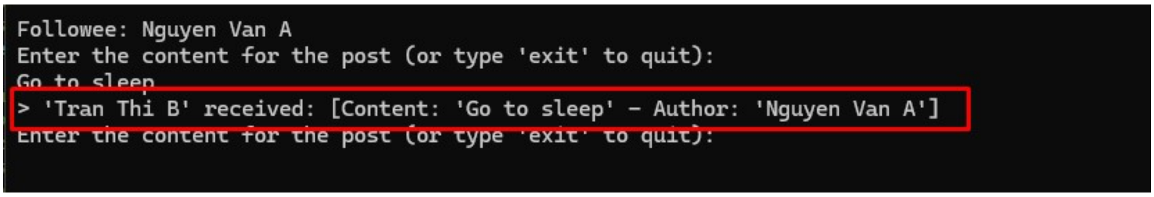
Exercise 3: Notification System with In-Memory Pub/Sub

Idea:

- Create 2 functions:
    - Subscribe: Register to follow other users to receive new information from them.
    - Publish: Post and push notifications to users who have subscribed.

Result: Users who have subscribed will receive notifications (in the code, it is hardcoded that "*Nguyen Van A*" is the followed user, and other users follow this user).

Here, "*Tran Thi B*" has followed "*Nguyen Van A*," so when *A* enters content, it will display a notification for *B* as:



```
Followee: Nguyen Van A
Enter the content for the post (or type 'exit' to quit):
Go to sleep
> 'Tran Thi B' received: [Content: 'Go to sleep' - Author: 'Nguyen Van A']
Enter the content for the post (or type 'exit' to quit):
```

Unit test:

- Currently, only 2 cases are being tested:
  - Receiving notifications from the followed user.
  - Not allowed to receive notifications from a user not followed.
- Since the Subscribe function from the beginning already takes in 1 delegate, when testing, you only need to set a flag to check whether the function is executed or not to test the 2 cases above.