## Kelsey Nash Design Feedback

Reviewer: Jose Montoro

Repo: <https://github.com/kelseynash7/SystemDesignAssign4/tree/main>

Feedback: I focused on the PersonalBlog class: <https://github.com/kelseynash7/SystemDesignAssign4/blob/main/personalBlog/src/main/java/com/blog/personalBlog/entity/PersonalBlog.java>

One area where PersonalBlog.java can be improved is the Single Responsibility Principle:

1. **Multiple Responsibilities**:
   * The class handles multiple concerns: user management (e.g., logIn, createUser), blog post management (e.g., createBlogEntry, readBlogEntry), and tracking user login status. This violates SRP, as it should only focus on one responsibility.
2. **Encapsulation and Cohesion**:
   * The PersonalBlog class tightly couples user management with blog management. Ideally, there should be separate classes for managing users and blog posts to enhance cohesion and encapsulation.

**Refactor Suggestions**

1. **Separate User and Blog Post Management**:
   * Create a UserService class responsible for user-related functions like logIn, createUser.
   * Create a BlogService class responsible for blog-related functions like createBlogEntry, readBlogEntry.
2. **Repository Pattern**:
   * I see that you’re using a Repository for `BlogRepository.java`, you could do the same for `User`.

One additional potential improvement is in `BlogController.java`: <https://github.com/kelseynash7/SystemDesignAssign4/blob/main/personalBlog/src/main/java/com/blog/personalBlog/controller/BlogController.java>  
  
Currently, there's no **error handling** for cases like when a post is not found. Implementing exception handling, possibly with @ControllerAdvice, would improve robustness and user feedback.

Hope that helps!

## Coder/Programmer Response

Thank you for the feedback. I think there may have been some confusion on what code was in use at this time. This is my fault - all methods in the unused classes are commented with //TO DO and only return a brand new object so I thought it was clear, but I should have either commented more clearly or just removed the code for now. It looks like you’ve only recommended one change based on SOLID principles when it should have been three. Also the class you commented on is not actually in use at this time. While I do intend to extend/refactor to more match my initial plan from our first assignment in the class, I haven’t had the time. And while I do appreciate that you reviewed other areas of my code, the suggestions aren’t really within the scope of this assignment.

**\*Note for Akib/Pawan** - Usually I would send this feedback back to Jose so I actually have three recommendations based on class (SOLID principles). However, I do not have time. I received this feedback Saturday afternoon and am currently at a conference so only have a few hours tonight (Saturday) and maybe tomorrow (Sunday) to accept/deny and make a change based on one of the recommendations, so I do not feel that I can do this. Unfortunately that also means that every recommendation made above is not actually on code that’s in use.

As the PersonalBlog.java class is not actually being used, I have to reject the recommendation that it should be refactored to separate responsibilities via the Single-Responsibility Principle. I have since removed that class. If it was in use, I would absolutely agree that having blog post and user methods in the same class is not ideal and violates the principle and I would implement changes to separate these responsibilities.

I do however agree with the refactor suggestions - they were future plans anyways, just not necessarily a part of this assignment. I have added Repository, Controller, and Service classes for Users and Categories - making sure that they follow the Single-Responsibility Principle by only having each deal with its specific model type (Post, User, Category).