2-

it needs some parts fixed as it violates the solid principles;

SRP: In this diagram we have the single responsibility principle. All the classes are responsible for one entity and not more.

OCP: The design is open to extensions but we’ve got to change some of the methods invisible for other classes.

LSP: Subtypes are substitutable for their base classes. All the Functions that use pointers or references to base classes are able to use objects of derived classes without knowing them.

ISP: there are no unused methods implemented.

DIP: in our class diagram some of high level modules are dependent on low level modules, like the Task class being dependent on tag class.

3-

Coupling:

Interaction coupling;

No direct coupling: none of the methods calls the other one.

Inheritance coupling:

We have a high level of inheritance.

Cohesion:

Method Cohesion:

Here, most of the methods perform a single problem-related task, some of them like ‘answer to daily questions’ in User class use the output of another method (“answer” in “Question”). Some methods like (“edit user profile” in User class) is actually dependent on the variable we pass to it (“which profile items we want to edit”). So the cohesion is of the Logical type.

Class Cohesion:

Some of our classes have only one visible method to outside of class like Question and Tag, which doesn’t make sense.

Each visible method doesn’t perform one thing necessarily and we don’t have functional cohesion.

Here we have Mixed-role type because the all the classes have some attributes that have nothing to do with the underlying semantics of that class.