Lab C Review Form – ENSF380 W2024

This form was completed by group: 24
The reviewed UML design was done by group: 25.

Instructions:

- **Submit a PDF file consisting of this document (Save as PDF).**
- The submitted document must have your group number. For example, if you are in group 15, the document should be named 15.pdf.
- **This document will be shared with the group whose UML diagram was reviewed. Do not include any names or student IDs in this document.**
- You must provide the kind of feedback you would like to receive - helpful, accurate, and appropriate. Failure to do so may affect your lab grade. If you receive feedback which does not meet these expectations, inform the instructor(s).

How to fill this form:

Follow the mentioned instructions and answer the following questions that are associated to a specific aspect of the UML design. Make sure that when answering each question, you consider the sub-questions provided with the bullet points. Submit a PDF file.

Reminder:

Refer to Lesson10 (UML) to help you in completing this design review process.

1. Structure and Clarity:

Q1.1: Is the structure of the diagram effectively organized to present the classes and their relationships in accordance with the UML notation?

- Is the diagram free of unnecessary clutter and complexity?
    - The UML diagram has the basic structure done well; at a glance it looks like a proper UML diagram. It has proper connections, but not too much. As will be described later, some more complexity is needed as specifics are lacking or improper.
- Does the diagram maintain a balance between detail and readability?
    - It is clear to see what each line is connecting without getting lost. The UML is well design in terms of readability and neatness. The blocks contain a small font size and have large white spaces which it difficult to read without zooming in. Altogether the structure is still clear and present.

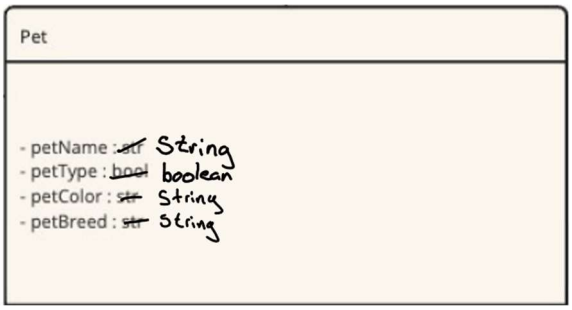
Q1.2: Are the labels and annotations in the diagram correct and reflect the logic of the narrative?

- Are the elements arranged in a way that the flow of data or control is intuitive?
    - It is difficult to see the flow of the chart, there does not appear to be a distinct organizational method. This can be rectified by placing the blocks in better locations, for example having classes that use / implement other class in a left to right top to bottom orientation. The arbitrary use of bidirectional arrows hinders the ability to see how data/control flows throughout. Relationships are distinct and cardinalities are clear. The clarity of the purpose of each class is also present through the description of the methods.

2. Classes, Methods, and Attributes:

Q2.1: Are the classes provided with a proper set of attributes that align with the requirement narrative?

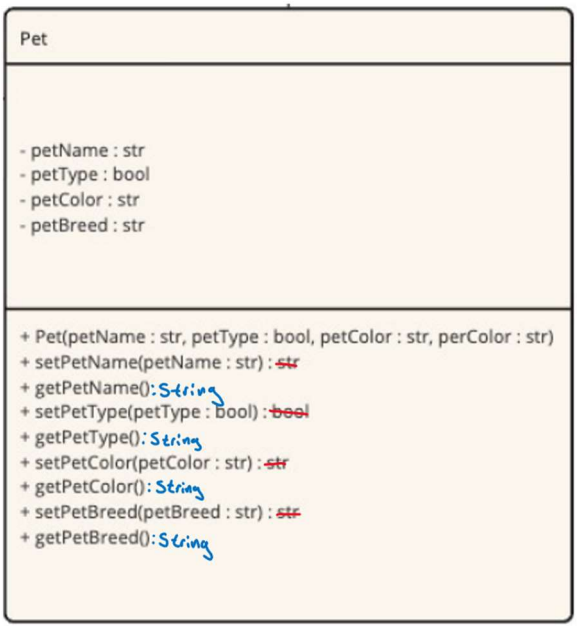
- Does the design show correct use of object-oriented principles, with each class having a clear, distinct purpose (encapsulation)?
  - The classes are structured to encompass the necessary attributes aligned with the narrative, such as a “Client” class containing name, phoneNumber, and Address. This alignment indicates a thoughtful consideration of the real-world entities and their characteristics that the software aims to model.
- Are the attributes have the proper data type?
  - The attribute The attributes are properly structured and relevant to their purpose, enhancing readability and maintainability. The choice of data types is appropriate but not consistent with Java syntax in same cases. For example:
    - Incorrect usage of boolean primitive: Use `boolean` instead of `bool`.
    - Wrapper for String: `str` instead of `String`.



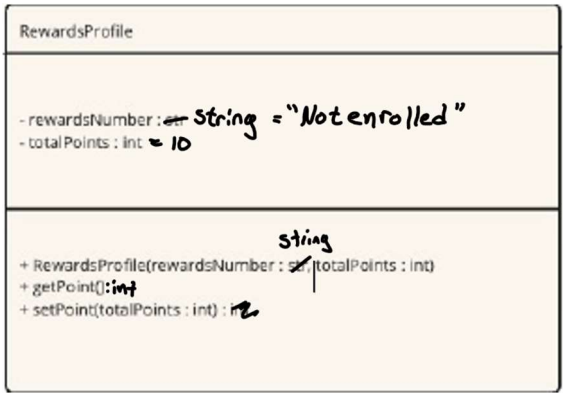
- Do the attributes have the proper access modifiers (private, public, and protected)?
  - The design correctly implements access modifiers (`private` and `public`), with the only exception being the methods of the Client class. With the other classes being so consistent it is assumed this was an overlooked typo.

- Are all attributes that are necessary for the class to fulfill its role present?
- Is it clear from the names how each concept relates to the narrative?
    - It is shown through the bidirectional arrows the intended connections between classes but the attributes and methods that would connect them properly are missing. This suggests a gap in the design's ability to fully represent the narrative's requirements. For example:
        - There is a connection between Client and Pet, but it would be helpful for example to have an attribute called owner of type Client. This would make the connect be “a Pet ‘has a’ owner”, the arrow would then be unidirectional. Appropriate setter, getter, and a constructor parameter would also be needed.
        - This is also similar for:
            - Client and RewardPoints classes.
            - Pet and CareProfile.
            - Pet and EmergVet.
            - Booking and Pet/Employee
        - There should also be an associated relationship between Booking, Employee and Pet.

    - There are misplaced return types with getters and setters in a couple different classes this can be confusing for anyone implementing the code from UML. Getter should have a return type and setter should typically only have a parameter.



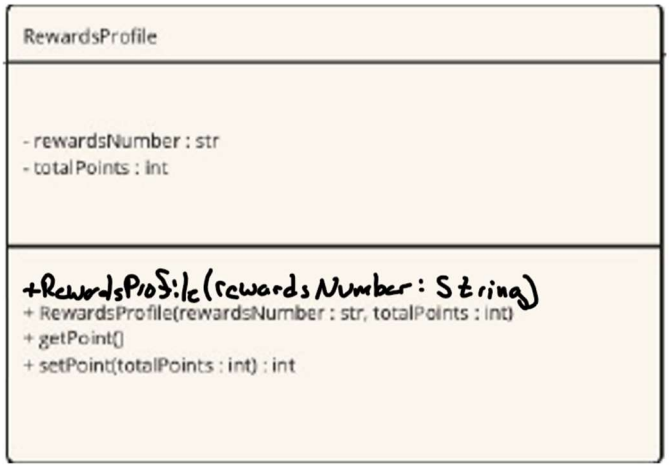
- Are all (non-final) variables written starting with a lowercase letter and in camel case?
  - Spelling and use of camel cases are well done and consistent across the diagram.
- Are default values specified for attributes where it makes sense, and are these defaults logical and clear?
  - Initialized values are not used, they would be appropriate to use in RewardsProfile.



Q2.3: Does each class have all the needed methods corresponding to the requirements narrative?

- The use of correct camel casing and well-defined method names contribute significantly to code readability and maintainability.

- Are constructors present for initializing the class's objects appropriately?
- Do constructors allow for different initialization scenarios (overloading) if needed?
  - Overloading is not used but not strictly needed. An example of where you could have used it is the constructor of the RewardsProfile class. You could have one constructor with an initial total point and one without.



- Are all (non-constructor) methods written starting with a lowercase letter and in camel case?
  - Very consistent across the diagram
- Do method names indicate all expected behavior?
  - As mentioned above, the lack of methods and attributes that could be used to properly connect Classes takes away from the narrative and makes it difficult to know what is going on.

3. Relationship Accuracy:

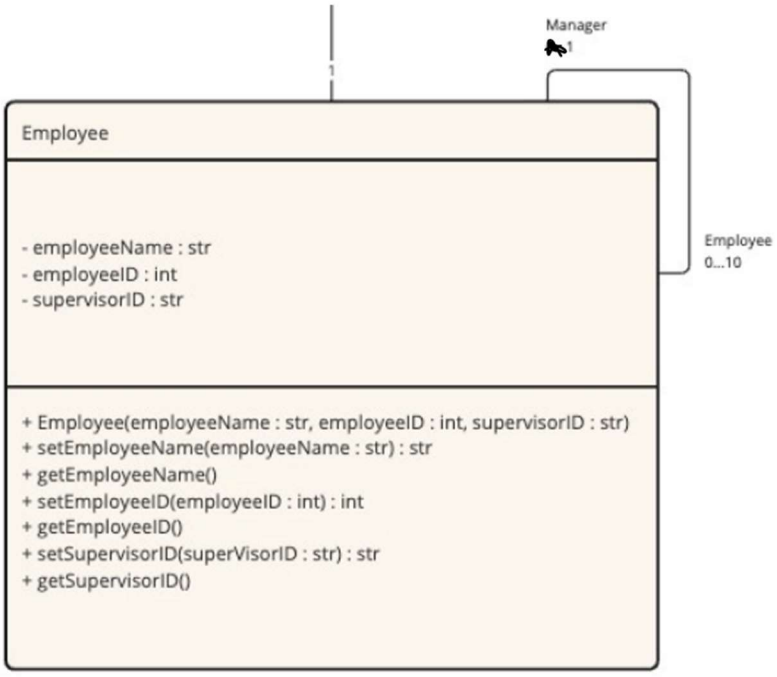
Q3.1: Are the interconnections between classes depicted correctly and do they logically arise from the narrative's context?

- Are associations between classes correctly identified and depicted in the diagram?
  - The relationships between classes are present through the solid arrows but incorrect use of arrow types to depict associations makes it difficult to follow the narrative.
- Do the associations (e.g., aggregation, composition) accurately reflect the nature of relationships (like "part-of" or "has-a") as per the narrative?
  - The “part-of” and “has-a” narrative is missing do to arrow usage.
  - Employee, Pet and Client classes have a combined arrow. This may have been used for neatness but incorrectly used in this context.
- Are dependencies between classes correctly shown?
- Only bidirectional arrows were used which makes it difficult to understand the narrative context and flow of the UML diagram. This does not indicate the correct dependencies.

- Are there redundant relationships that could be simplified or removed?
  - You did a good job keeping the UML simple there are no redundancies with methods or attributes but there are critical.
  - The Exception call is missing dotted lines and throw key word.

Q3.2: Are the cardinalities in the relationships applied correctly, and do they logically fit within the context of the requirements narrative?

- Are the cardinalities (like 1..1, 0..10, etc.) at each end of an association accurately represented?
    - Cardinalities are present and accurate, for employee to manager it should be ‘1 ‘rather than ‘0..1’.



- In most places, special cardinalities are handled well but in other places may need more detail,
- Good complex relationships

4. Consistency and Adherence to the narrative:

Q4.1: Does the diagram fully represent all the narrative's elements and there is no essential component is missed? (The UML should have 9 classes; names were provided in instructions)

- All the necessary classes, names are correct. No spelling errors.

Q4.2: Does the diagram consistently maintain a level of detail, ensuring a coherent application of elements and notation throughout?

- The level of detail is consistent for the UML diagram and would serve as a general outline but as mentioned there is a lack of detail.

Q4.3: Does the design accurately represent the specifications that provided in the narrative?

- It represents the specifications enough but not perfectly,

Q4.4: Are there component/element in the diagram that deviate from the narrative's details and may need further review? (This deviation might represent a contradiction to the specified requirements.)

- No components deviate at all, all components are accurate to the document. Although some things are missing.

5. Error Management and Code Efficiency:

Q5.1: Does the diagram clearly indicate the handling of the incorrect inputs or the unexpected user interactions?

Q5.2: Does the UML diagram provide a comprehensive and unambiguous design that can be directly translated into efficient and maintainable code structures?

- The design is efficient enough besides missing some of the crucial elements described before.

Q5.3: Are the given design ensure a logical sequence and simplify the complexity in future code implementation?

- Lacking logical sequence as mentioned before.

Overall feedback:
[Please provide overall feedback that could enhance the UML diagram.]

Overall, I think your group did a good job. The missing methods and attributes that connect the appropriate classes and the incorrect use of arrows had a cascading effect for this review and influenced all our feedback. Your group was very consistent with what you did do both correct and incorrect.

We suggest comparing your UML with model solution on D2L along with the suggestions made.