**Author:** Ralph Vitug

**Subject:** Intermediate Software Development

**Section:** COMP-2327 (268396)

**Topic:** Assignment 1 – Reflection

**Topic 1:** Benefit of class diagram

Class diagrams are very helpful because they give a visual representation of classes, attributes and methods, which makes it easier to understand the structure of the assignment before or while coding it. By laying out the design visually, I can see how different classes and methods are set or planned on how to are to be executed. This also promotes structure and a more organized way on laying out codes.

For example, assignment 1 was designing a banking system. The diagram provided showed the class Client, in which contained private attributes like client\_number, first\_name, last\_name and email address. Furthermore, <<stereotype>> is declared on the diagram to make said attributes available to the outside word as a read only. Additionally, the diagram also includes the class BankAccount, which similarly share the same idea with the class Clientwhere some attributes are public, and some are private. (diagram will indicate the sign “- “for private and “+” for public)

An example of a diagram is shown below.

|  |
| --- |
| **Client** |
| * Client\_number: int * First\_name: str * Last\_name: str |
| +\_\_init\_\_(client\_number: int, first\_name: str, last\_name: str)  +<<property>>client\_number : int  +<<property>>first\_name: str  +<<property>>last\_name: str  +\_\_str\_\_: str |

**Topic 2:** Test plan

A formal unit test plan provides the advantage of organization and completeness, since it promotes efficiency and accuracy of every test case. It pushed me to carefully think through every condition a method should handle before writing tests. For example, in the BanckAccount, the test plan(.xlsx) required verifying not only valid transactions but also invalid transactions, such as, non numeric inputs, negative inputs, exceeding limits, etc. Another benefit is that test plans make collaboration easier, since teammate can clearly see what has been tested and what remains. However, a disadvantage is that creating detailed plans can feel time consuming and redundant, especially for smaller projects where the stress may outweigh the benefits.

Example of a test plan:

|  |  |  |  |
| --- | --- | --- | --- |
| Condition | Precondition | Method | Result |
| Returns first\_name | First\_name=”Wendy” | None | First\_name=”Wendy” |
| Exception raised when first\_name is blank | None | First\_name=” “  Last\_name=”Ways” | ValueError |
| Returns string expected format | Ways, Wendy | None | Ways, Wendy |