SIT771 Object-Oriented Development

Pass Task 6.1: Document the Account Design

Focus

Make the most of this task by focusing on the following:

- Concept:

  Focus on deepening your understanding of Universal Modeling Language (UML) and its various diagram types, emphasizing its effectiveness and usage in enhancing communication, design, and overall efficiency in the programming process.

Overview

In this task, you will be approaching the documentation of the design of the Bank Program. You'll create *Universal Modeling Language (UML)* diagrams to visualize the design of the current state of the bank program.

Submission Details

Submit the following files to OnTrack.

- An image of your UML class diagram. Have its orientation as portrait, even if it is a landscape image. OnTrack will convert it to fit on an A4 page.
- A UML sequence diagram of the Transfer

Spend some time reflecting on what you have learnt, focus on the concepts and ideas, how they work, and how they are related. You want to demonstrate that you have really understood these ideas and why an approach to program documentation is important.

Instructions

You need to create the following:

- A UML Class Diagram showing the static structure of the program
- A UML Sequence Diagram showing how the Transfer works.

Creating a Class Diagram

Using a tool such as (the free) LucidChart, create a UML class diagram representing the current state of the Banking program which you have been creating throughout the unit. Ensure that you show all of the classes, their fields, properties, and methods.

Here are some quick pointers to help get you started with this task:

- In Lucid Chart, you can create a new Blank UML document. When creating a new document, select **UML** from the Template categories. Then select **Blank UML**.

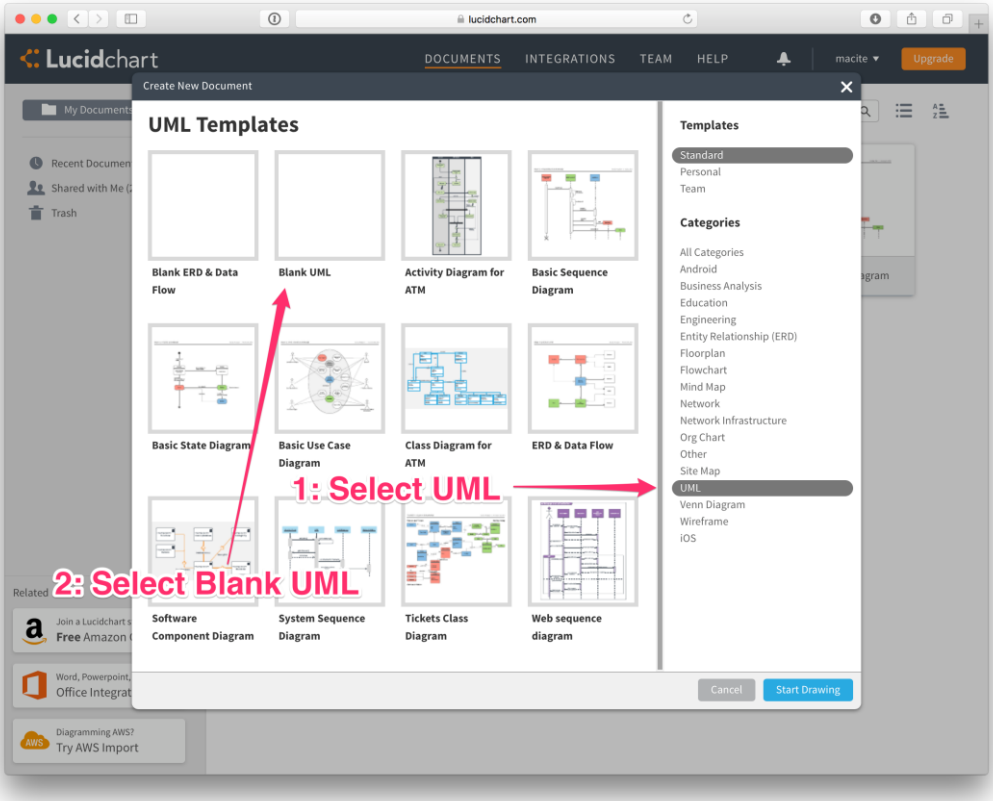

Figure: Select a new Blank UML document

- To add a class, locate the **UML Class Dgm** shape pallet on the left, then drag across a class (its the first one on the left).

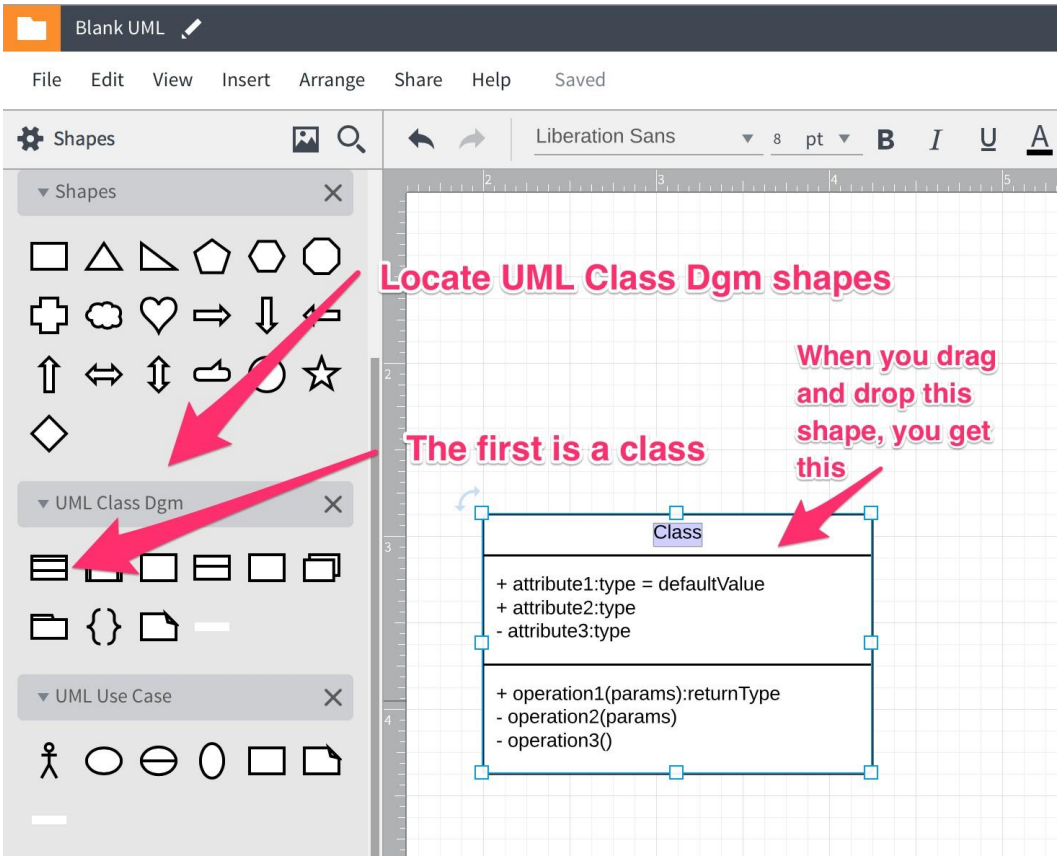

Figure: Drag in a class shape

- Edit the **class name** in the top box

- Adjust the fields and properties in the 2nd box

    - Use `+` for public `-` for private members
    - Private fields are written `- fieldName: Type`
    - For properties, add a property or readonly property stereotype. For example `+ Name: String << property >>`.
- Include methods and constructors in the 3rd box
    - Use `+` for public `-` for private members
    - Public methods are written `+ methodName ( param1: Type, param2: Type) : Type`
    - Static methods should be underlined

Preparing your submission

Once you have the diagrams finished, you can download your work as an image. From the **File** menu in LucidChart (not your browser) choose **Download As**. In the settings choose the following options.

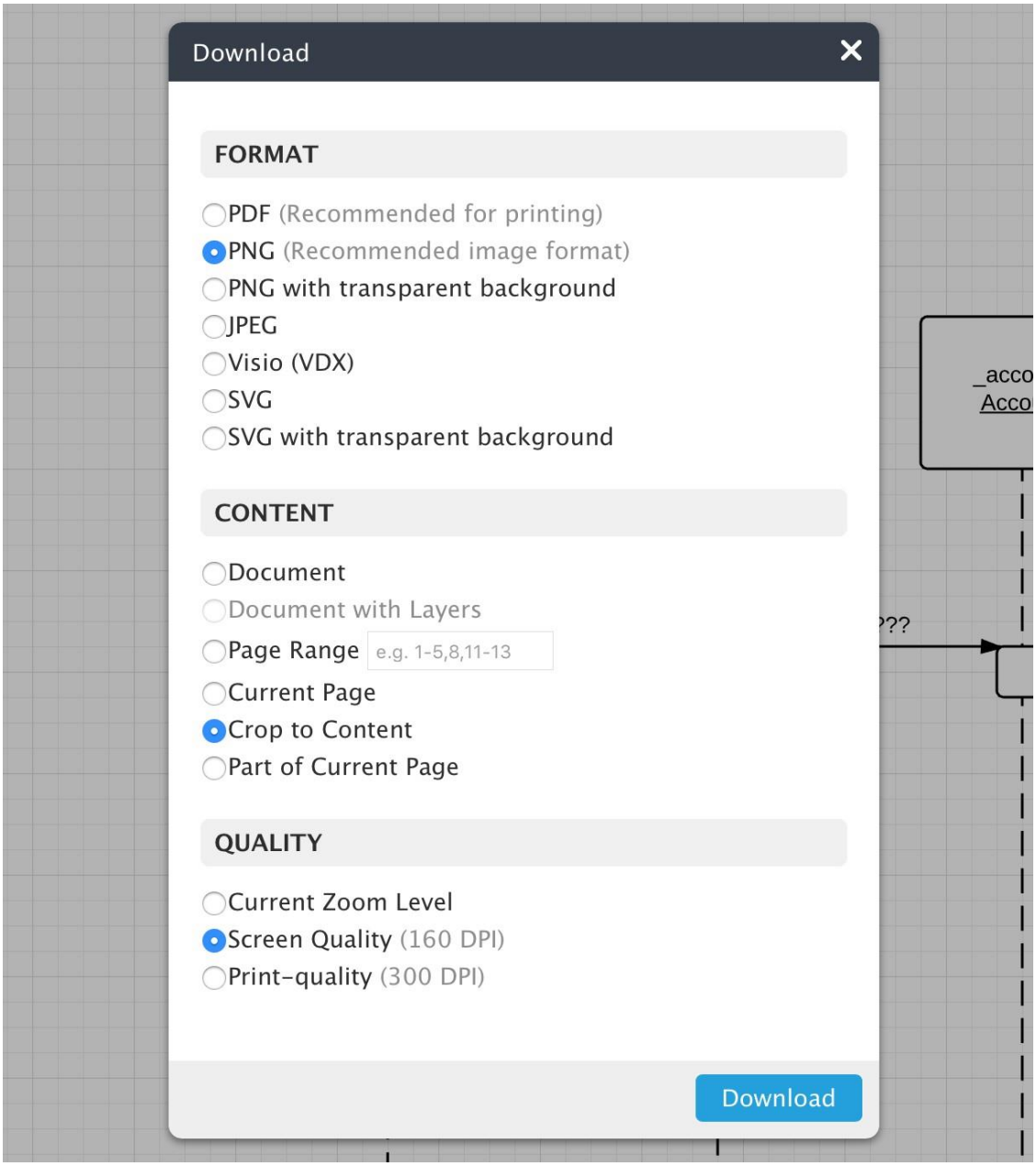

Figure: Download settings

When you hit download it will take a few seconds, and you will have the image downloaded to your

machine. Save these images somewhere and upload them to OnTrack.

Remember to keep a copy of these images in your backup.

Task Discussion

For this task you need to discuss the use of UML to communicate the static structure of amn object oriented solution:

- How you represent classes, fields, methods, relationships
- Describe the relationship between the diagram and your code
- How could you use this to think through a solution **before** you write the code? What would be the advantage to doing this?