PBL4: Team-based Creative Design G12-2024

# **UML Guidelines**

#### 1. Use Descriptive Names for Classes and Members:

Ensure that class names, attributes, and methods clearly describe their function. This improves clarity and makes the diagram more understandable.

### 2. Show Relationships Clearly:

Depict associations, aggregations, compositions, and dependencies between classes with appropriate lines and symbols. Use arrows and multiplicity notations where necessary.

### 3. Indicate Access Modifiers:

Use standard UML visibility indicators (+ for public, - for private, # for protected) for all class members to clearly communicate their accessibility.

### 4. Use Consistent Diagram Layouts:

Maintain a consistent style across your diagrams in terms of font sizes, line styles, shapes, and spacing. A visually cohesive diagram improves readability.

# 5. Avoid Overcrowding:

Avoid cluttering the diagram with too many details. If needed, break down complex systems into smaller, more focused diagrams to improve clarity.

# 6. Update Diagrams with Code Changes:

Ensure the UML diagram reflects any updates or changes in the code. Keeping diagrams up to date helps in maintaining an accurate system overview.

# 7. Label Associations and Define Multiplicities:

Clearly label associations between classes and specify multiplicities (e.g., 1..\*, 0..1) to accurately depict the relationships between different elements.

#### 8. Document Changes with Date and Author:

For each update to the diagram, include a note with the name of the person who made the change and the date. This helps in tracking modifications and versioning.

#### 9. Include Key Attributes and Methods Only:

For clarity, include only the most important attributes and methods in the class diagram. Avoid listing trivial or excessively detailed elements unless they are crucial to understanding the system, which keeps the diagram concise and focused on core functionality.

# 10. Use Notes to Clarify Complex Logic:

For classes or relationships that involve complex logic or specific constraints (e.g., business rules or data validation), include UML notes attached to the relevant elements. This helps provide additional context and understanding without cluttering the diagram itself.