Practical - 02

Subject: Object Oriented Modelling and design (OOMD)

Assignment Title: Class diagrams for capturing and Structuring System design.
Semester: 7th, Fourth Year.
Instructor: Saiprasad Bhise.
Submission Date: 01/09/2025

1] Objective

- Understand and apply standard UML guidelines for modelling system structure using Class diagrams.
- Create accurate, professional Class digrams that include attributes, methods, and relationships.
- Use clear class templates to communicate system design to varied stakeholders.

2] Problem Statement:

Draw one or more class diagrams that represent the system structure for a chosen application. Complement these diagrams with detailed Class templates describing main classes - attributes, operations, relationships and design rationale

Smart Automation System. - A system that allows residents to control and monitor home devices (lights, fans, security cameras) using a controller.

3] Introduction to class diagram Modelling -

- Class diagrams are fundamental UML artefact. They illustrate classes, attributes, methods and relationships such as association, aggregation, composition and inheritance between objects in a system.
- Class diagrams clarify the static structure and help define how different entities interact throughout the software's life cycle. They are essential for transforming requirements into actionable insights and facilitate communication between analysts, designers and developers

4] THEORY AND BEST PRACTICES -

* UML Elements -

- Class - Represents a blueprint with attributes (states) and methods (behaviour).
- Attributes - Properties / data fields.
- Methods - (Operations) - Functions performed by the class.
- Relationships - Association, Aggregation, composition, Inheritance, dependency.

* Naming notation -

- Class names : nouns , capitalized (eg. Customer)
- Attributes / methods : descriptive, consistent ; visibility (+ public, - private, # protected).
- UML notation - Rectangle with three sections.
  - Top : Class Name.
  - Middle: Attributes.
  - Bottom: Methods.

- Diagram Layout:
  - Centralize core classes, place supporting ones around.
  - Draw relationships using correct UML symbols (lines/arrows, diamonds for aggregation / composition)
  - Indicate multiplicity (eg. * 1) and roles.

- Class Template structure :
  - Class Name
  - Description.
  - Attributes (name, type, visibility)
  - Methods (name, parameters, return type, visibility).
  - Relationships (type, target, classes, multiplicity).
  - Constraints and business rules.
  - Notes / assumptions.

5] Assignment workflow -

1. System definition and boundary.
   State what your system does. outline main components and boundaries.

2. Identify major classes -
List core entities, their roles and interactions.

3. List attributes and methods.
For each class, identify state (attributes) and behaviour

4. Define relationships:
Specify association, aggregations, compositions, inheritance and dependencies.

5. Draw class diagrams -
Use proper UML notation and layout:

6. Develop class Templates.
For at least two key classes, document detailed templates as per structure above.

7. Stakeholder validation (Hypothetical) -
Describe how feedback / refinement would be gathered to ensure design meets needs.

6] Deliverables:

- Title page with assignment metadata.
- Introduction to class diagram modelling (standards)
- Clearly labelled class diagrams in UML notation.
- Completed class templates for minimum two major classes.
- Optional stakeholder validation/ reflection notes.
- Typed, well formatted document (PDF/ hard copy).

7] Evaluation criteria. -

- Accurate use of UML notation and diagram clarity.
- Professional, consistent naming and layout.
- completeness of class templates - attributes, methods relationships and rules.
- Application of best industry practices.
- presentation quality and timely submission.

8] Recommended tools -

- UML Tools - Draw IO, Microsoft Visio, Lucidchart, Astah, etc
- Collaborative review tools - Miro, confluence.

9] System description -

The proped system is smart automation designed to provide centralized monitoring and control of household devices. It allows homeowners to alltomate and remotely manage appliances such as lights, fans, air conditioners, security cameras and door locks through a mobile app.
The system integrates sensors (motions, temperatures, smoke) and connected devices to improve energy efficiency, safety, and convenience.

10] Assignment requirements -

- System boundary - Clear defined and reasonable in scope.
- Classes - Well identified, appropriate responsibilities.
- Relationships - proper associations, inheritance, aggregation etc. with correct notation and multiplicities.

- Class diagram - Standard, precise and clearly structured.
- Templates - Detailed description of all essential elements per template guide.
- Stakeholder validation - Reflection or sample feedback notes.
- Documentation - professional and readable formatting.

11] Conclusion -

Class diagram modelling is a key step bridging requirements and design. Standardized diagrams and class templates ensure clear understanding of how system entities interact, support stakeholder communication and lay the groundwork for robust software construction - skills required for real world object oriented development.

* * * * * * *