

Milestone 1 Template — Requirements & Initial Class Diagram

Document Title: Milestone 1 – Requirements Analysis, Decomposition & Abstraction

1. Project Title

(Write your project name)

template: *Library Management System*

2. Problem Description (5–7 sentences)

Explain briefly:

- What the system will do
- Who will use it
- Why it is needed

Template:

“In this project, I aim to design a system that manages ... The main users will be ... The system will help solve the problem of ... The key functions include ... This project will apply OOP and Computational Thinking to decompose the problem and build an efficient solution.”

3. Functional Requirements (User Stories)

List at least 5.

Format:

- “As a **(user type)**, I want to **(action)** so that **(purpose)**.”

Example:

- As a librarian, I want to add new books so that the library database stays updated.

4. System Decomposition (CT: Decomposition)

Break the system into **main modules or components**.

Template:

“The system can be divided into the following modules:

1. _____
2. _____
3. _____

Each module has a clear responsibility in the system.”

5. Identified Classes (CT: Abstraction)

List at least 3–6 classes.

Example Template Table:

Class Name	Attributes	Methods
_____	_____	_____
_____	_____	_____
_____	_____	_____

6. Relationships between real-world objects

(Students can attach a photo, draw using draw.io, or insert a screenshot.)

7. Encapsulation Evidence (Code Snippets)

Include short code samples showing:

- private attributes
- getter/setter
- constructors

8. Reflection (4–6 sentences)

Explain what you learned about:

- Decomposing a problem
- Identifying classes

Template:

“In this milestone, I learned that decomposition helps me break down complex tasks into manageable parts. I also practiced abstraction by identifying the essential classes needed in the system. Describe relationships between classes/objects. This analysis will guide the next milestone where I implement encapsulation.”

Milestone 2 Template — Encapsulation & Basic Operators

Document Title: Milestone 2 – Pattern Recognition

1. Summary of Progress (3–5 sentences)

Template:

“In this milestone, most objects require Create, Read (display information), Update, and Delete (CRUD) operations and search information. As a result, the project should use Arrays of Objects or ArrayList to manage collections of objects.”

2. Implemented management Classes Overview

(List the management classes you created and basic operators. Student use pseudo code or flowcharts to describe algorithms).

template:

Name class: BookList

***Description:** The BookList class is implemented in the project to manage the collection of books in the library. This class provides basic operations such as adding new books, updating book information, deleting books, and searching for books based on their identification codes*

Basic operator specifications:

Add a new book :

Begin

Step 1: the user inputs the information of book such as : id, title,...

Step2: validate input data

Step3:

End

3. Encapsulation Evidence (Code Snippets)

(Include short code samples showing)

4. Explanation of Abstraction / Pattern Recognition

Template:

“I noticed that several classes share common method patterns such as create(), update(),.... Based on these patterns, I designed my methods more consistently. This helped improve the clarity and reusability of the code.”

5. Challenges and Solutions

Template:

“One challenge I faced was... I solved it by... This helped me better understand encapsulation.”

6. Reflection (4–6 sentences)

(Write what you learned from implementing encapsulation and pattern recognition.)

Milestone 3 Template — Inheritance & Polymorphism

Document Title: Milestone 3 - Hierarchy & Polymorphism,

- Generalization (Advanced Abstraction)

1. Summary of New Additions

Template:

“In this milestone, I created a class hierarchy using inheritance and implemented polymorphism by overriding methods. This helps the system handle different types of objects using a common interface.”

2. Class Hierarchy Design (CT: Generalization)

Describe your parent and child classes.

Template:

“The parent class is _____.

The child classes are _____.

The shared attributes include _____.

Each subclass adds unique behaviors such as _____.”

3. UML Hierarchy Diagram

Insert your updated UML diagram with inheritance arrows.

4. Polymorphism Demonstration (Code Snippet)

Show overridden methods + dynamic dispatch.

Template:

```
Book a = new TextBook();  
a.displayInfor(); // polymorphism
```

Then explain how polymorphism works in your project.

5. Algorithmic Thinking Evidence

Explain the logic behind method overriding.

Template:

“I designed the overridden methods by analyzing how each subclass behaves differently. This required algorithmic thinking to ensure that each version of the method follows the correct sequence of steps.”

6. Reflection (4–6 sentences)

What you learned about inheritance and polymorphism.

Milestone 4 Template — Final Submission, Evaluation & Reflection

Document Title: Milestone 4 - File I/O

- Final System Integration & CT Reflection

1. System Overview and Integration

Provide a brief overview of the completed system.

Describe how different components (basic classes, manager classes, menu system) work together.

Explain how File I/O is integrated into the existing OOP-based design.

Describe Computational Thinking (CT) Reflection.

2. Updated UML Class Diagram (Final Version)

Insert diagram.

3. OOP Concepts Application Summary

Explain how each OOP concept appears in the system:

- **Encapsulation:**
- **Inheritance:**
- **Polymorphism:**
- **Abstraction:**

4. CT Skills Evaluation

Students must indicate where Decomposition was applied and identify which parts of the code demonstrate Pattern Recognition.

Decomposition

Abstraction

Pattern Recognition

Generalization

Algorithmic Thinking

Evaluation & Iteration

Template example:

“I applied decomposition when I separated the system into modules such as...

I used abstraction by focusing on key attributes...

I applied iteration when I reviewed and improved my class diagram...”

5. System Testing & Evidence

- Screenshots
- Sample inputs/outputs
- Explanation of how you tested (3–4 sentences)

6. Self-Evaluation (Strengths & Limitations)

Template:

“One strength of my system is...

However, a limitation is...

In the future, I can improve by...”

7. Final Reflection (6–8 sentences)

Focus on:

- What you learned from OOP
- What CT skills improved
- What you would do differently next time