UML Master Cheat Sheet 2026

The 8 essential UML diagrams every BSIT student needs to defend a capstone

itsourcecode.com

Published by PIES Information Technology Solutions

Compiled from Nym's archive of 370+ walkthroughs

August 2026 Edition

UML Master Cheat Sheet 2026

The 8 essential UML diagrams every BSIT student needs to defend a capstone.

One printable reference covering what each diagram is, when to use it, the notation you must know, and where to find working examples on itsourcecode.com. Compiled from Nym's 370+ diagram walkthroughs, tuned for Filipino BSIT and international CS students shipping capstone documentation in 2026.

How to use this cheat sheet

1. Look up the diagram type you need in the table of contents.
2. Read the "When to use it" line to confirm the fit.
3. Study the notation table to make sure you use the correct symbols.
4. Follow the link to a full walkthrough on itsourcecode.com for a real-system example.
5. Adapt the example to your own system and defend it in front of the panel.

Print this reference and keep it next to you during Chapter 3 documentation.

Table of contents

Structural diagrams

`Class Diagram`

What it is: A static view of the classes in a system, their attributes, methods, and the relationships between them (inheritance, association, aggregation, composition, dependency).

When to use it: Every capstone documentation (Chapter 3). Any object-oriented codebase needs one to explain the domain model.

Key notation:

Element	Symbol	Purpose
Class	Rectangle with 3 compartments	Name / attributes / methods
Visibility `+`	Public	Accessible from anywhere
Visibility `-`	Private	Class-internal only
Visibility `#`	Protected	Class + subclasses
Inheritance	Solid line + hollow triangle	Subclass -> Superclass
Association	Solid line	Two classes linked
Aggregation	Solid line + hollow diamond	"has-a", parts can outlive whole
Composition	Solid line + filled diamond	"part-of", parts die with whole
Multiplicity	`1`, `0..1`, `1..*`, `*`	How many instances relate

Common capstone systems: School Management, Library, Inventory, POS, Hospital, HRIS.

See real examples:

- Student Course Registration System Class Diagram
- School Management System Class Diagram
- Class Diagram for Restaurant Management System
- Full archive (32 examples): /topics/class/

Component Diagram

What it is: Shows the physical software components that make up a system, their interfaces, and how they connect. Higher level than a class diagram.

When to use it: When your capstone has multiple deployable pieces (web app, mobile app, background worker, third-party API, database). Panels expect it for any distributed or multi-tier system.

Key notation:

Element	Symbol	Purpose
Component	Rectangle with two smaller rectangles on the left edge	A deployable software unit
Provided interface	Circle on a line ("lollipop")	Interface the component offers
Required interface	Half-circle on a line ("socket")	Interface the component needs
Assembly connector	Ball-and-socket joining lollipop + socket	Two components wired together
Dependency	Dashed arrow	Component uses another
Port	Small square on component edge	Distinct connection point

Common capstone systems: Railway Reservation, Airline Reservation, Food Ordering, LMS, Point of Sale.

See real examples:

- Component Diagram for Railway Reservation System
- Component Diagram for Airline Reservation System
- Online Food Ordering System Component Diagram
- Full archive (29 examples): /topics/components-diagram/

Deployment Diagram

What it is: Shows the physical hardware (nodes) that run the software and how software artifacts are distributed across them. This is your infrastructure map.

When to use it: When your capstone runs on more than one machine (web server + database server), when you deploy to cloud, or when your defense panel asks "where does this actually run?"

Key notation:

Element	Symbol	Purpose
Node	3D box	A physical or virtual machine
Artifact	Rectangle with `<<artifact>>` stereotype	A deployable file (jar, war, .exe, container image)
Communication path	Line between nodes	Network link, sometimes labeled with protocol
Device node	3D box with `<<device>>`	Physical hardware
Execution environment	3D box nested inside a node	OS, JVM, runtime container
Deployment	Dashed arrow with `<<deploy>>`	Artifact is placed on a node

Common capstone systems: Online Shopping, Credit Card Processing, Blood Bank, LMS, Any web app with database backend.

See real examples:

- Deployment Diagram for Online Shopping Cart
- Deployment Diagram for Credit Card Processing System
- Deployment Diagram for Blood Bank Management System
- Full archive (26 examples): /topics/deployment-diagram/

ER Diagram (Entity-Relationship)

What it is: A model of the data your system stores: what entities exist, what attributes describe them, and how they relate. This is the visual blueprint for your database schema.

When to use it: Every capstone with a database. Panels will ask for the ER diagram before they even look at your code. Pair it with your SQL schema in Chapter 3.

Key notation (Chen notation and crow's foot):

Element	Chen symbol	Crow's foot	Purpose
Entity	Rectangle	Rectangle with entity name	A thing you store data about (User, Order, Product)
Weak entity	Double rectangle	Rectangle with double border	Depends on another entity for identity
Attribute	Oval	Column inside entity box	A property of an entity
Primary key	Underlined attribute	`PK` marker	Unique identifier
Foreign key	Attribute connected to another entity	`FK` marker	Link to another entity
Relationship	Diamond between entities	Line between entities	How entities connect
Cardinality `1:1`	`1` and `1` on relationship	Single line + single line	One-to-one
Cardinality `1:M`	`1` and `N`	Single line + crow's foot	One-to-many
Cardinality `M:N`	`M` and `N`	Crow's foot + crow's foot	Many-to-many (needs junction table)

Common capstone systems: Hospital, Hotel Reservation, Food Ordering, School Management, Library, E-commerce.

See real examples:

- ER Diagram for Hospital Management System with SQL Schema
- ER Diagram for Online Hotel Reservation System
- ER Diagram for Online Food Ordering System
- Full archive (39 examples): /topics/uml/erd/

Behavioral diagrams

`Use Case Diagram`

What it is: Shows the actors who interact with the system and the use cases (goals) each actor pursues. This is the highest-level view of what the system does from the outside.

When to use it: Chapter 1 or Chapter 3 of every capstone. Answers "who uses this and what can they do?"

Key notation:

Element	Symbol	Purpose
Actor	Stick figure	External user or system that interacts
Use case	Oval	A goal or task the system supports
System boundary	Rectangle around use cases	The scope of what your system covers
Association	Solid line	Actor uses this use case
Include	Dashed arrow with `<<include>>`	Use case ALWAYS uses another (required sub-step)
Extend	Dashed arrow with `<<extend>>`	Use case OPTIONALLY uses another (conditional)
Generalization	Solid line + hollow triangle	Sub-actor inherits from super-actor

Common capstone systems: Every capstone has one. Airline Reservation, Employee Attendance, Library, Hospital, LMS.

See real examples:

- Use Case Diagram for Airline Reservation System
- UML Use Case Diagram Tutorial with Examples
- Use Case Diagram for Employee Attendance Management System
- Full archive (43 examples): /topics/uml/usecase/

Sequence Diagram

What it is: Shows how objects interact over TIME. Reads top-to-bottom (time) and left-to-right (participants). Shows the exact order of messages between objects to accomplish a single use case.

When to use it: When you need to prove a specific scenario works. Panels love sequence diagrams for login flows, checkout flows, and any multi-step process.

Key notation:

Element	Symbol	Purpose
Actor / Object	Rectangle at top of lifeline	A participant in the interaction
Lifeline	Dashed vertical line down from object	Represents the object over time
Activation	Thin rectangle on lifeline	Object is executing during this time
Synchronous message	Solid arrow with filled arrowhead	Caller waits for response
Asynchronous message	Solid arrow with open arrowhead	Caller does not wait
Return	Dashed arrow	Response from callee back to caller
Self-message	Arrow looping back to same lifeline	Object calls its own method
Alt / Opt / Loop	Rectangle "frame" with label	Alternative paths, optional steps, loops

Common capstone systems: Banking login, Attendance check-in, Hostel booking, E-commerce checkout, Payroll processing.

See real examples:

- Sequence Diagram for Online Banking System
- Attendance Management System Sequence Diagram
- Sequence Diagram for Hostel Management System
- Full archive (36 examples): /topics/uml/sequence/

Activity Diagram

What it is: A flowchart-like view of the workflow through a system. Shows the order of activities and decision branches for a single business process.

When to use it: For any workflow that has branches, loops, or parallel activities. Chapter 3 for order processing, checkout, admission, appointment, or any multi-step business process.

Key notation:

Element	Symbol	Purpose
Initial node	Filled black circle	Starting point
Final node	Filled black circle inside larger circle	End point
Activity	Rounded rectangle	A single step or action
Decision	Diamond	Branching point (yes / no or multiple paths)
Merge	Diamond (multiple in, one out)	Rejoin after branching
Fork	Thick horizontal or vertical bar (one in, many out)	Start of parallel activities
Join	Thick bar (many in, one out)	Wait for all parallel activities to finish
Swimlane	Vertical column with actor label	Group activities by responsible actor
Object flow	Dashed arrow with object	Data flowing between activities

Common capstone systems: E-commerce checkout, Attendance clock-in, Payroll process, Admission, Order fulfillment.

See real examples:

- Activity Diagram for E-Commerce Checkout
- Activity Diagram for Attendance Management System
- Activity Diagram for Payroll Management System
- Full archive (34 examples): /topics/uml/activity/

Data Flow Diagram (DFD)

What it is: Not strictly UML but universally required in Filipino BSIT capstones. Shows how data moves through the system: from external sources, through processes, into data stores, and back out to sinks.

When to use it: Chapter 3 of every BSIT capstone. Panels expect Level 0 (context), Level 1 (major processes), and often Level 2 (decomposition of one Level 1 process).

Key notation (Yourdon-DeMarco):

Element	Symbol	Purpose
External entity	Rectangle	Person, department, or external system sending / receiving data
Process	Circle (or rounded rectangle in Gane-Sarson)	Transforms input data into output data. Numbered (1.0, 2.0, 1.1, 1.2)
Data store	Open-ended rectangle (two horizontal lines)	Where data rests (a table, file, or persistent store)
Data flow	Arrow with label	Named data moving between elements
Level 0 (Context)	One process representing entire system	Highest-level overview
Level 1	3 to 7 processes decomposed from Level 0	Major system processes
Level 2	Decomposition of a single Level 1 process	Detail for one specific area

Common capstone systems: Inventory, Hotel, Hospital, School, Library, POS, E-commerce.

See real examples:

- DFD for Inventory Management System (Levels 0, 1, 2)
- DFD for Hotel Management System
- Hospital Management System DFD 2026 All Levels
- Full archive (43 examples): /topics/uml/dfd/

Panel defense tips

Six things Filipino BSIT panels ask about UML diagrams (2026 edition):

1. **"Explain your class diagram in your own words."** Do not read attribute names off the slide. Talk about the domain: what are the main things your system manages, how do they relate?
2. **"Why did you use inheritance here?"** Have a real reason. "Because both classes are users" is weak. "Because Admin extends User with additional permissions" is strong.
3. **"What actors appear in your use case diagram?"** Know all of them. Missing an obvious actor (like the database admin, or the payment gateway as an external system) gets flagged fast.

4. **"Walk me through your sequence diagram for use case X."** Practice narrating your happy-path sequence diagrams out loud before defense. Panels want to hear you follow the arrows.
5. **"Show me your ER diagram."** Have SQL schema next to it. Panels test whether the diagram matches the actual database.
6. **"What is the difference between DFD and activity diagram?"** DFD shows DATA movement. Activity diagram shows WORK flow. Panels ask this to check you didn't just copy an activity diagram and label it DFD.

Get the matching source code

Every capstone system in this cheat sheet also has working source code on itsourcecode.com in one or more of PHP, Python, Java, VB.NET, C#, and JavaScript. Search by system name at /topics/free-projects/ or browse by language:

- PHP Projects (319 capstones)
- Java Projects (127 capstones)
- VB.NET Projects (296 capstones)
- Python Projects (259 capstones)
- JavaScript Projects (55 capstones)

About this cheat sheet

Compiled 2026-08-18 by the itsourcecode.com editorial team from Nym's archive of 370+ UML diagram walkthroughs. Every diagram type includes a working example plus the full walkthrough link so you can adapt real systems to your own capstone. Panel defense tips are based on questions BSIT panels in the Philippines actually ask during 2024-2026 defenses.

For the full archive of 306+ UML examples across all diagram types:

- UML Diagrams Library
- Use Case Diagram archive
- Sequence Diagram archive
- Activity Diagram archive
- ER Diagram archive
- DFD archive