

Session 2

Use case diagrams

A use case diagram shows what the system does for each actor without saying how. It is the diagram examiners use to check that your scope statement and your later sequence diagrams agree.

Objectives

✕ Do not copy. Read for understanding and the viva

- Complete questions 4 to 5 of the manual: use case diagrams
- Prepare the deliverable before the lab and finish it during the session
- Be ready to explain every step in the viva

Questions Covered

✕ Do not copy. Read for understanding and the viva

Question	Requirement	Status
Q4	Draw Use Case Diagram for Online Shopping System	Complete
Q5	Draw Use Case Diagram for Library Management System	Complete

Preparation

✕ Do not copy. Read for understanding and the viva

- The manual asks for a problem description of 300 to 500 words and a list of assumptions before every diagram. Write both first; they fix the scope the evaluator marks you against.
- Identify primary actors (who gets value) and secondary actors (external systems such as a payment gateway).

- Name use cases as verb phrases: Place Order, Cancel Order. Use include and extend only where the manual's text justifies them.
- Draw the system boundary and keep actors outside it.

Question 4

Problem Statement

■ Write in lab record

Draw Use Case Diagram for Online Shopping System.

Solution

■ Write in lab record

Assumptions

The Online Shopping System is the same single-retailer web store modelled in Session 1. This diagram records what the system does for whom, not how it is built, so the classes of Session 1 do not appear; only the goals of the people and external systems that touch the store.

Two kinds of people use the system. A Customer is anybody who buys. A new visitor registers with name, email, phone and password. A registered customer logs in, browses the catalogue by category, searches products by name, and adds or removes products in a personal shopping cart. From the cart the customer places an order: the system asks for a delivery address, takes payment and confirms the order with an order number. After that the customer can track the order and, until it has been shipped, cancel it for a refund. The customer can also maintain saved addresses. An Administrator is the retailer's staff member who keeps the catalogue current (add, edit and remove products, update stock), views incoming orders and marks each paid order as shipped with a courier and tracking number.

Two external systems take part. The Payment Gateway authorises card and UPI payments and processes refunds; the store never handles card numbers itself. The Courier Service receives shipment details and returns tracking updates. Both are secondary actors: they do not start any use case, they respond to the system.

Placing an order always requires the customer to be logged in and always involves a payment step, so Place Order includes Login and Make Payment. Cancel Order includes Refund Payment for the same reason. Applying a promotional code is optional and happens only when the customer types one at checkout, so Apply Coupon extends Place Order at that extension point. Login is also included by Manage Cart, Track Order and Cancel Order. Product reviews, wish lists, returns after delivery, multiple sellers and administrator sales reports are out of scope.

Assumptions:

- One primary actor per goal: the customer buys, the administrator sells. There is no separate delivery agent user.
- Browsing and searching do not need login; everything that touches a cart or order does.
- Payment is always taken at order time; cash on delivery is treated as a payment mode inside Make Payment.
- The Payment Gateway and Courier Service are secondary actors drawn on the right of the system boundary.
- Include is used for behaviour that always happens; extend for behaviour that happens only under a condition (coupon entered).

Steps

1. Model, Add Diagram, Use Case Diagram; name it `OnlineShoppingUseCases`.
2. Toolbox Use Case Subject: draw one rectangle and name it `Online Shopping System`. Every use case must sit inside it.
3. Toolbox Actor: place Customer and Administrator on the left, Payment Gateway and Courier Service on the right, all outside the rectangle.
4. Toolbox Use Case: add the fourteen ovals from the table inside the rectangle.
5. Toolbox Association: click an actor, drag to each of its use cases (plain line, no arrow head).
6. Toolbox Include: click Place Order, drag to Login; the tool draws a dashed arrow labelled include. Repeat for the include rows. Toolbox Extend: click Apply Coupon, drag to Place Order (arrow points at the base use case).
7. Save and export as PNG.

Diagram

Actor	Kind	Goal
Customer	Primary	Find products, buy them, follow and cancel orders
Administrator	Primary	Keep the catalogue and stock correct, ship paid orders
Payment Gateway	Secondary (external system)	Authorise payments and process refunds
Courier Service	Secondary (external system)	Accept shipments and report tracking status

Use case	Actors	Relationship
Register	Customer	none
Login	Customer, Administrator	included by Manage Cart, Place Order, Track Order, Cancel Order, Manage Products, Ship Order
Browse Catalogue	Customer	none
Search Product	Customer	none
Manage Cart	Customer	includes Login
Place Order	Customer, Payment Gateway	includes Login, Make Payment; extended by Apply Coupon
Make Payment	Payment Gateway	included by Place Order
Apply Coupon	Customer	extends Place Order (condition: coupon code entered)
Track Order	Customer, Courier Service	includes Login
Cancel Order	Customer, Payment Gateway	includes Login, Refund Payment
Refund Payment	Payment Gateway	included by Cancel Order
Manage Addresses	Customer	includes Login
Manage Products	Administrator	includes Login

Use case	Actors	Relationship
Ship Order	Administrator, Courier Service	includes Login

■ Lab record: every tab is one file of the answer. Write all of them.

Sketch

```

1          +-----+
2          |          Online Shopping System          |
3          |                                          |
4  Customer -----|--( Register )                    |
5          |                                          |
6  +-----|--( Browse Catalogue )    ( Search Product )--+
7          |                                          |
8  +-----|--( Manage Cart )-----<<include>>-----+
9          |                                          v
10 +-----|--( Place Order )-----<<include>>--( Login )-----|
11 |          |      ^      |                      ^  ^
12 |          |      |      +---<<include>>--( Make Payment )-----|
13 |          |      <<extend>>                      | |
14 |          |      |                      | |
15 +-----|--( Apply Coupon )                      | |
16 |          |                      | |
17 +-----|--( Track Order )-----<<include>>-----+
18 |          |                      |
19 +-----|--( Cancel Order )---<<include>>--( Refund Payment )---|
20 |          |                      |
21 +-----|--( Manage Addresses )                    |
22 |          |                      |
23 |          ( Manage Products )-----|
24 |          |                      |
25 |          ( Ship Order )-----+-----|
26 |          |                      | |
27 +-----+-----|-----+
28 |          |                      |
29 Courier Service Pay
30 (also Track Order)

```

Explanation

Each oval is a goal an actor can state in one verb phrase, which is the test for a use case; “Cart” alone would be a class, “Manage Cart” is a use case. Login is included rather than drawn as a precondition on every oval because the manual’s examples use include for shared mandatory steps, and it keeps the diagram honest: Place Order cannot complete without it. The extend arrow points from Apply Coupon to Place Order, the opposite direction to include, because the base use case is complete without the extension. The two secondary actors sit on the right and connect only to the use cases that call them; a common error is to draw the gateway on Place Order itself, which hides the fact that payment is a reusable step. Place Order is the use case traced by the Session 3 sequence diagram.

Question 5

Problem Statement

■ Write in lab record

Draw Use Case Diagram for Library Management System.

Solution

■ Write in lab record

Assumptions

The Library Management System runs the lending desk of a university library. It follows the manual’s Figure 1.5, which shows a Student and a Librarian with Borrow Book and Return Book, and grows it to the full set of goals the library needs.

A Member is any student or staff member with a valid library card. Members search the catalogue by title, author or ISBN and see how many copies are available. A member can reserve a title whose copies are all issued; the first reserver is told when a copy comes back. At the desk, a member borrows a copy by showing the card: the system checks that the card is valid, that the member has not reached the borrowing limit and that the copy is not reserved for someone else. Returning a copy closes the loan; if it is overdue, the system calculates the fine and the member pays it before the return is accepted. A member with no overdue copies can renew a loan once online. Members can also view their own loans and fines.

A Librarian issues and receives books on behalf of members at the desk, registers new members and renews or cancels cards, adds new copies when a purchase arrives, removes lost or damaged copies, and collects fines. The Librarian also prints daily reports of issued, overdue and reserved books. An Administrator creates librarian accounts and sets the policy values: loan period, borrowing limit and fine per day.

Every member action that touches a loan needs a valid card, so Borrow Book, Return Book, Renew Book and Reserve Book include Verify Membership. Return Book is extended by Pay Fine only when the copy is overdue. Borrow Book is extended by Notify Reserver only when the returned copy has a pending reservation. Purchasing decisions, inter-library loans, e-books, and digital payment of fines are out of scope; fines are paid in cash at the desk.

Assumptions:

- Two primary human actors, Member and Librarian, plus an Administrator for setup; no external systems.
- Borrow, Return and Renew are started by the Librarian at the desk for a Member; Search, Reserve, Renew and View Loans are also available to the Member online.
- A member can hold at most three copies and renew each loan once.
- Fine is a fixed amount per day and is always paid in full before a return is accepted.
- Reports are read only and produced from the loan records; no separate reporting system.

Steps

1. Model, Add Diagram, Use Case Diagram; name it `LibraryUseCases`.
2. Toolbox Use Case Subject: draw the boundary `Library Management System`.
3. Toolbox Actor: place Member and Librarian on the left, Administrator on the right.
4. Toolbox Use Case: add the fourteen ovals from the table inside the boundary.
5. Toolbox Association: connect every actor to its use cases.
6. Toolbox Include: Borrow Book to Verify Membership and the other include rows. Toolbox Extend: Pay Fine to Return Book, Notify Reserver to Return Book. Set the extension point names in the Editor pane (`overdue` , `reserved`).
7. Save and export as PNG.

Diagram

Actor	Kind	Goal
Member	Primary	Find, reserve, borrow, renew and return books; pay fines
Librarian	Primary	Run the desk: issue and receive copies, manage members and copies, print reports
Administrator	Primary (setup)	Manage librarian accounts and lending policy

Use case	Actors	Relationship
Search Catalogue	Member, Librarian	none
Reserve Book	Member	includes Verify Membership
Borrow Book	Member, Librarian	includes Verify Membership
Return Book	Member, Librarian	includes Verify Membership; extended by Pay Fine (overdue), Notify Reserver (reserved)
Renew Book	Member, Librarian	includes Verify Membership
Pay Fine	Member, Librarian	extends Return Book
Notify Reserver	none (system)	extends Return Book
Verify Membership	none (system)	included by Reserve, Borrow, Return, Renew
View Loans and Fines	Member	includes Verify Membership
Register Member	Librarian	none
Manage Copies	Librarian	none
Generate Reports	Librarian	none
Manage Librarians	Administrator	none
Set Lending Policy	Administrator	none

■ Lab record: every tab is one file of the answer. Write all of them.

Sketch

```

1          +-----+
2          |          Library Management System          |
3          |
4  Member  -----|--( Search Catalogue )-----|
5          |          |
6          +-----|--( Reserve Book )-----<<include>>----+
7          |          |                                v
8          +-----|--( Borrow Book )-----<<include>>--( Verify Membership )--|
9          |          |                                ^   ^
10         +-----|--( Return Book )-----<<include>>----+ |
11         |          |      ^      ^                      |
12         |          | <<extend>> <<extend>>                  |
13         |          |      |      |                      |
14         +-----|--( Pay Fine ) ( Notify Reserver )      |
15         |          |                                |
16         +-----|--( Renew Book )-----<<include>>-----+
17         |          |                                |
18         +-----|--( View Loans and Fines )--<<include>>---+
19         |
20         | ( Register Member )-----|
21         | ( Manage Copies )-----|
22         | ( Generate Reports )-----|
23         |
24         | ( Manage Librarians )-----|
25         | ( Set Lending Policy )-----|
26         +-----+

```

Explanation

The manual's Figure 1.5 has two actors and two use cases; this diagram keeps those four elements and adds the rest of the desk's work so that the boundary is complete. Verify Membership is a system-internal use case: no actor is connected to it, it exists only because four other use cases share the same mandatory check. Pay Fine and Notify Reserver are extensions because a normal return needs neither; the extension point names (`overdue` , `reserved`) state the condition. Member and Librarian both connect to Borrow Book because either can start it: the member online, the librarian at the desk. The Administrator is kept as a separate actor even though a librarian might do the job, because the goals (policy and accounts) are different from

running the desk. If you later draw a sequence diagram for this system, trace Borrow Book: it has the most objects involved (Member, Card, Copy, Loan, Reservation).

Viva Questions

× Do not copy. Read for understanding and the viva

Q: What is an actor? **A:** A role played by a person or an external system that interacts with the system to achieve a goal. Actors are outside the boundary; the customer is an actor, the shopping cart is not.

Q: Difference between a primary and a secondary actor? **A:** A primary actor starts a use case to get value (Customer). A secondary actor is called by the system during a use case (Payment Gateway). Draw primary on the left and secondary on the right.

Q: When do you use include and when extend? **A:** Include for behaviour that always happens and is shared (Place Order includes Make Payment). Extend for optional behaviour under a condition (Apply Coupon extends Place Order only when a code is entered). Include arrows point at the included use case; extend arrows point at the base.

Q: Can a use case have no actor? **A:** Yes, if it is only ever included or extends another (Verify Membership). It must still be reachable through some use case that has an actor.

Q: Why is the system boundary important? **A:** It separates what the system does from what the actors do. Anything inside is a responsibility you must design; anything outside is assumed to exist.

Q: Is Login a good use case? **A:** On its own it gives the actor no business value, so purists leave it out. The manual's style accepts it as an included use case because it is a mandatory shared step; state which convention you follow.

Q: How does this diagram connect to the class diagram of Session 1? **A:** Each use case is realised by a set of collaborating objects from the class diagram; the sequence diagram in Session 3 shows that realisation for Place Order.

Common Mistakes

× Do not copy. Read for understanding and the viva

- Drawing use cases as steps of a process (Enter Address, Click Pay) instead of complete goals; each oval must deliver value on its own.
- Pointing the extend arrow from the base use case to the extension; extend and include arrows point in opposite directions.
- Putting an actor inside the system boundary, or drawing a database or the shopping cart as an actor.
- Connecting the Payment Gateway directly to Place Order and also to Make Payment; connect it once, to the use case that calls it.
- Adding arrowheads to actor associations; those are plain lines. Only include, extend and generalisation have heads.
- Forgetting the 300 to 500 word description; the evaluator uses it to decide whether your actors and use cases are complete.

Session Summary

■ Write in lab record

- Question 4: Online Shopping use case diagram, four actors (two primary, two secondary), fourteen use cases, ten include and one extend relationship
- Question 5: Library Management use case diagram, three actors, fourteen use cases, five include and two extend relationships, grown from the manual's Figure 1.5
- Actor and use case tables, ASCII sketch and PlantUML file for both diagrams
- Place Order is the use case traced in Session 3

[✎ Edit this page](#)

[< Previous](#)
Session 1

Session 3 [Next >](#)