FARABI UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

Lecture 12.
Integrated UML Modeling

Final case study and course review

Dr. Bazargul Matkerim, Professor-Researcher

Final lecture outcomes

Integrated modeling

Select diagrams for a specific question.

Trace a requirement across views.

Check names, multiplicities and behavior for consistency.

Course completion

Explain the library-loan model.

Identify missing and conflicting rules.

Derive tests from scenarios.

Present a coherent model portfolio.

Case study: a library borrowing system

Actors and goals

A librarian records borrowing and returns.

Members borrow available book copies.

The system records loans and due dates.

Scope and assumptions

One loan refers to one member and one copy.

A copy has at most one active loan.

Historical loans remain stored.

Reservations and fines are outside this case.

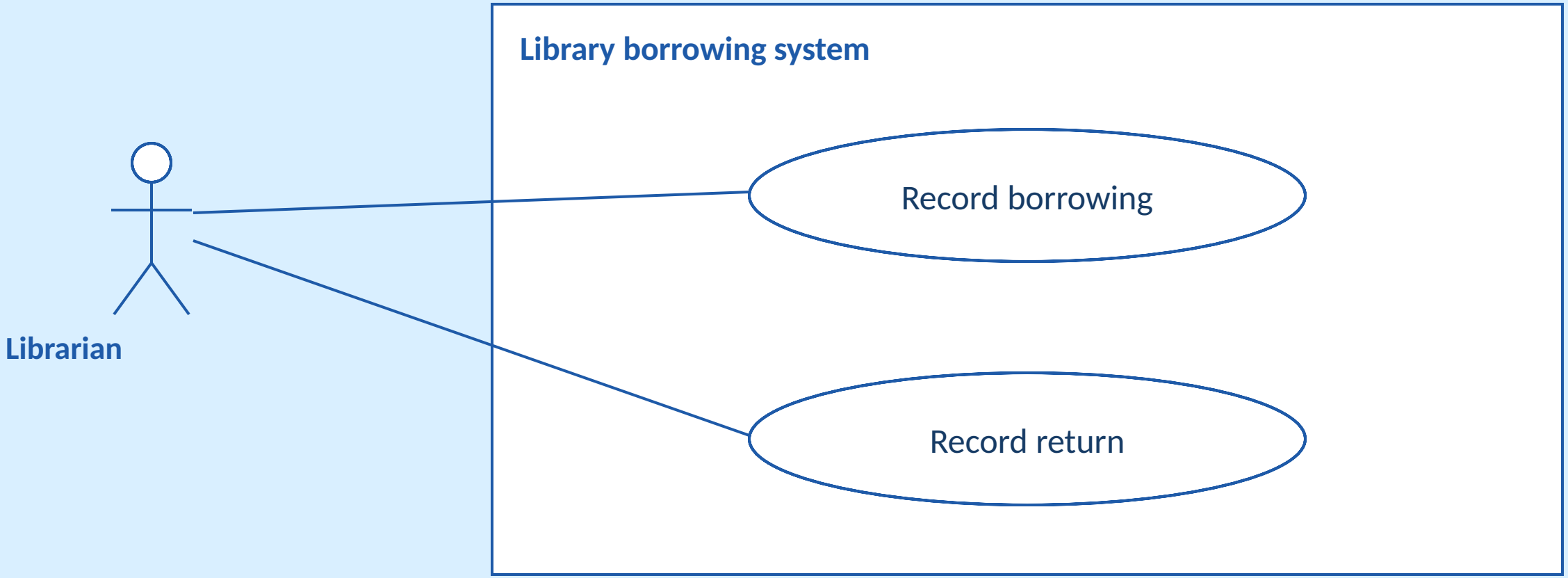
Requirements and business rules

ID	Requirement or rule
R1	An eligible member can borrow an available copy.
R2	A successful borrow creates one active loan and due date.
R3	A return closes the active loan and makes the copy available.
R4	A failed borrow leaves loan and copy data unchanged.
R5	A copy has no more than one active loan.

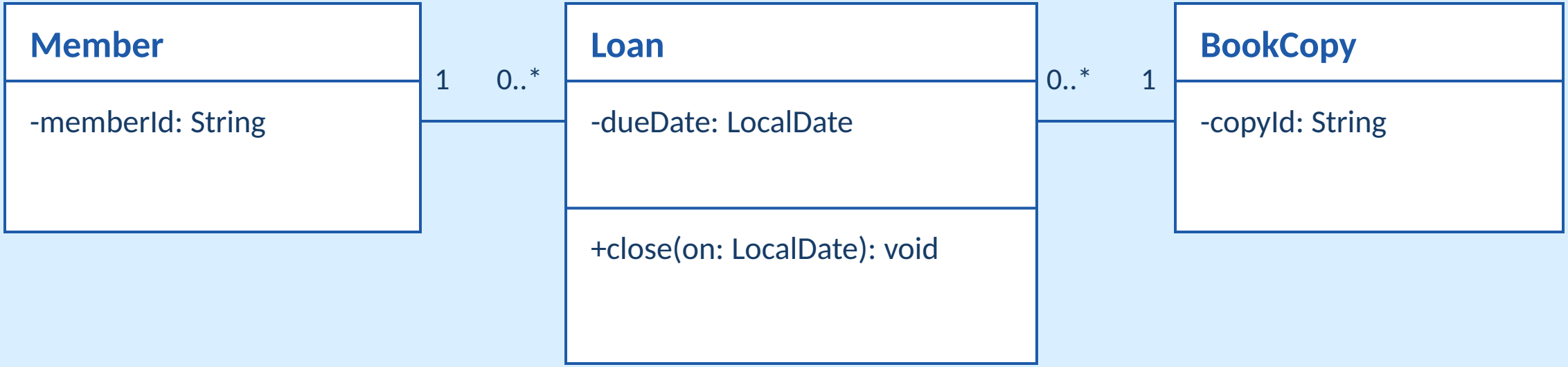
Choosing a UML view

Question	Suitable view
Who needs which capability?	Use case
What concepts and links exist?	Class and object
How do participants collaborate?	Sequence or communication
What work and lifecycle changes occur?	Activity and state machine
How is the solution organized?	Component, package and deployment

Use cases define system goals

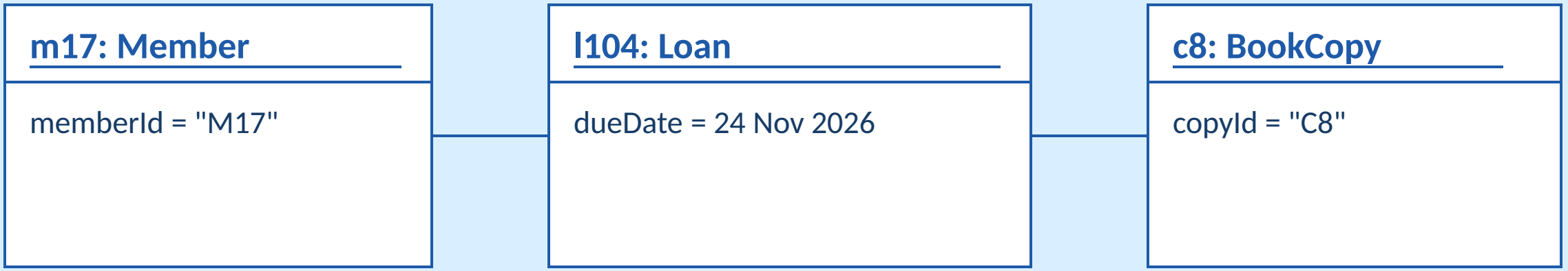


Class model: concepts and multiplicities



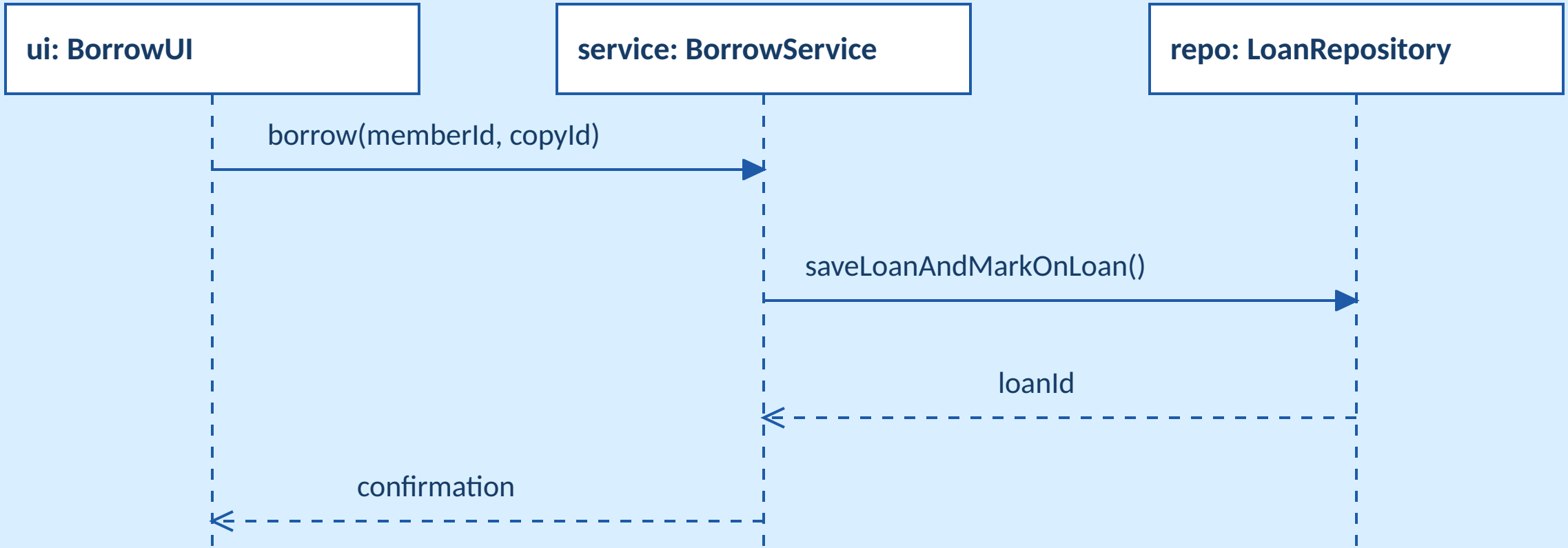
Constraint: at most one active Loan per BookCopy. Historical loans can be many.

Object diagram: one borrowing snapshot

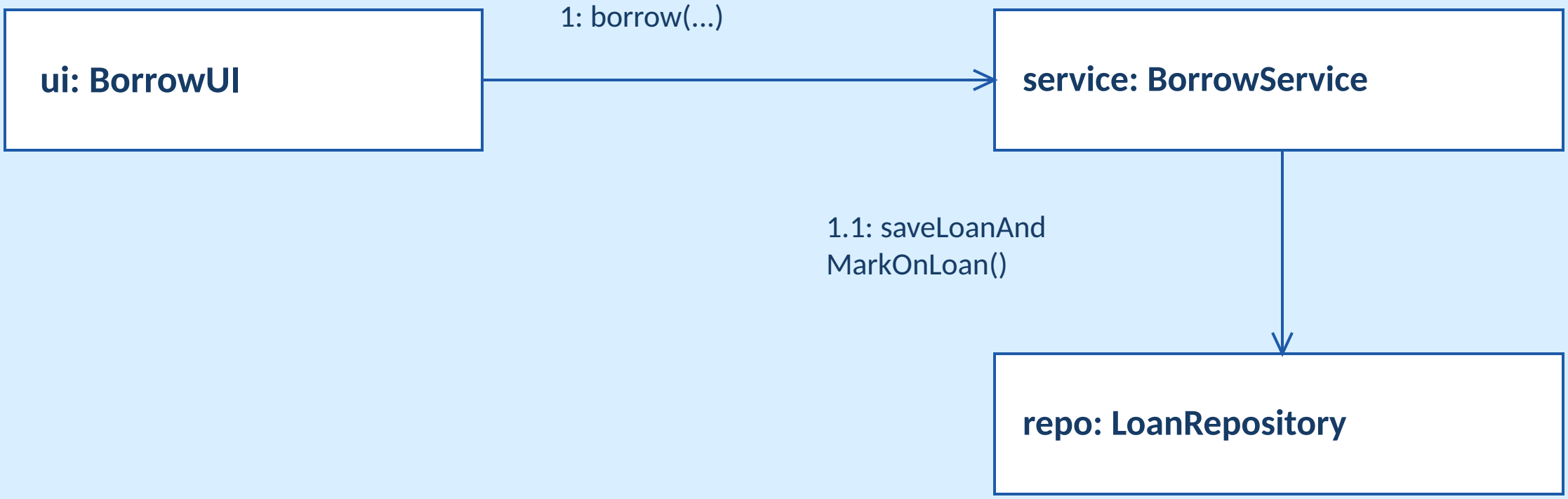


The object snapshot illustrates one valid instance configuration of the class model.

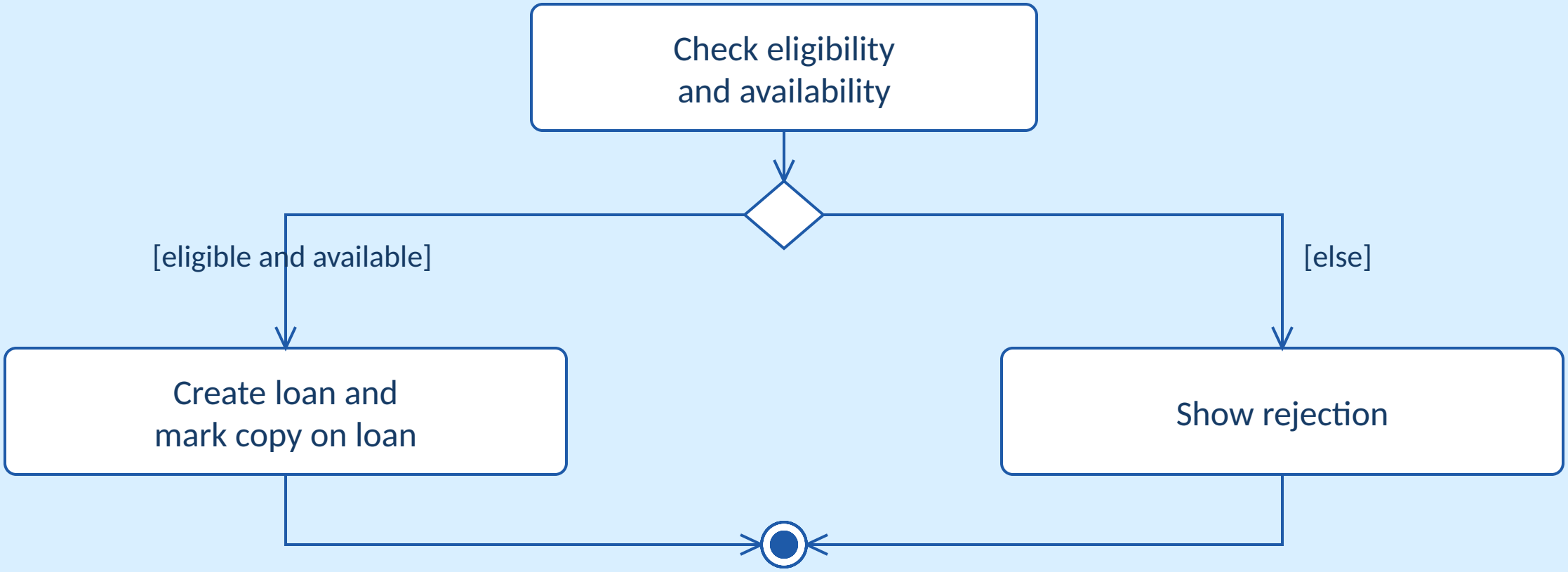
Sequence diagram: successful borrowing



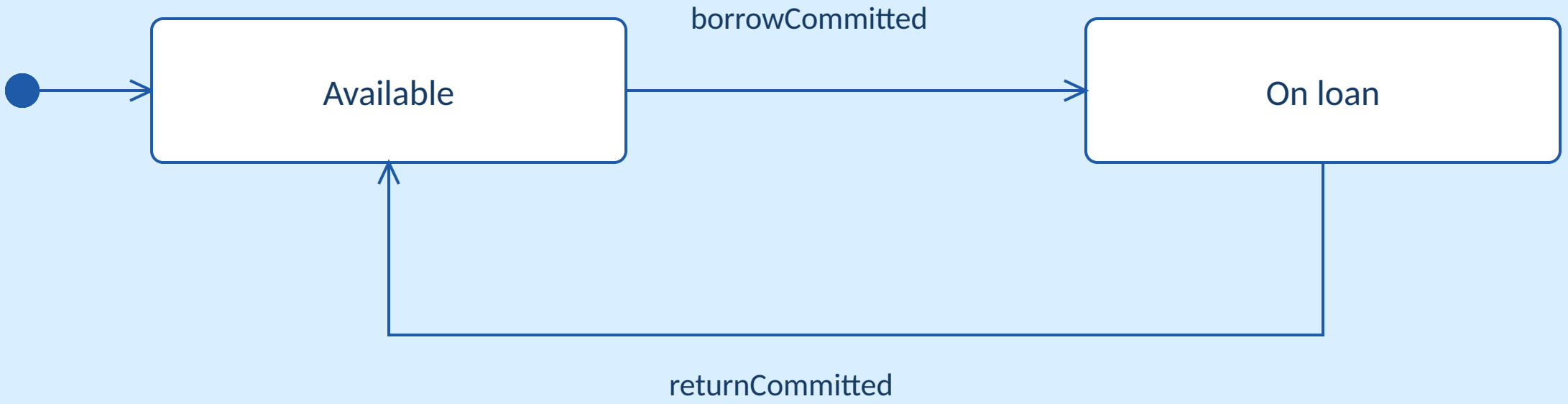
Communication view of the same collaboration



Activity model: borrowing alternatives



State model: the BookCopy lifecycle



Committed business outcomes trigger lifecycle changes in this model.

Component model: services and contracts



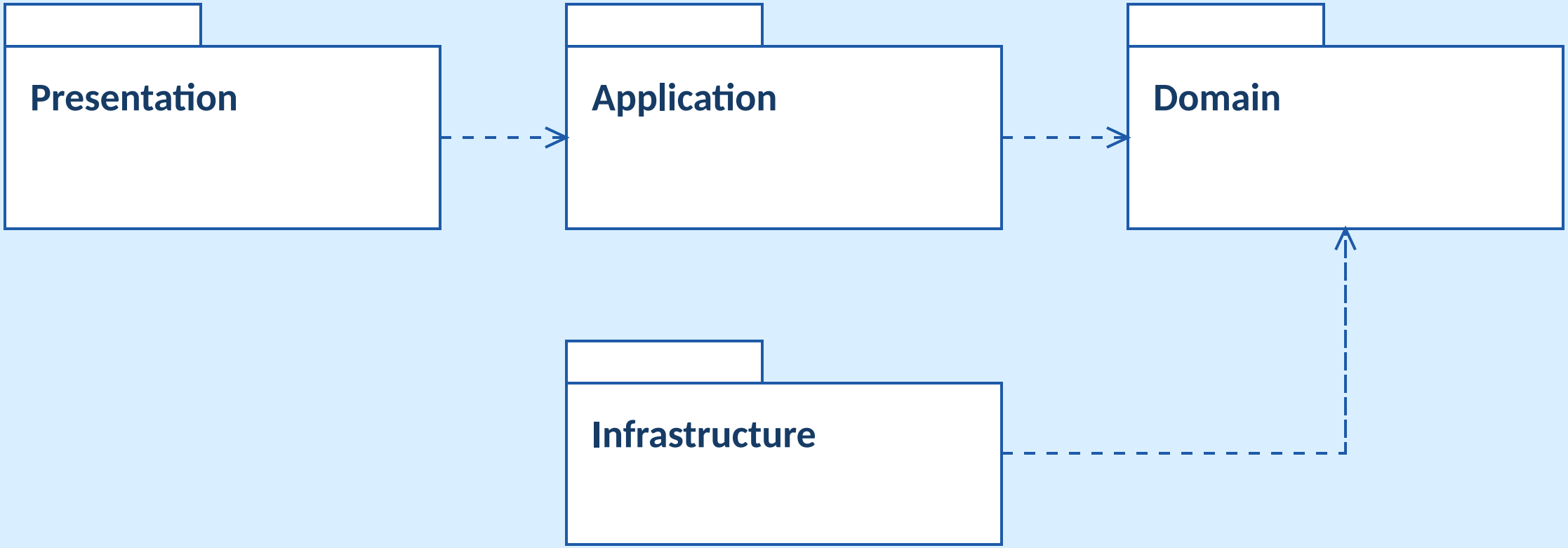
IBorrowing

UI requires it.
Loan Service provides it.

ILoanStore

Loan Service requires it.
Persistence provides it.

Package model: stable dependency direction



Domain owns the business concepts and repository abstractions in this design.

Deployment considerations

Possible deployment

Client device runs the librarian UI.

Application server hosts the loan service.

A database stores members, copies and loans.

Decisions to document

Which artifact runs on which node?

How do clients authenticate?

Where is transaction atomicity enforced?

What happens during a network failure?

Cross-diagram consistency checks

Check	Evidence across views
Names	Member, BookCopy and Loan denote the same concepts.
Operations	Interaction messages have defined receiving responsibilities.
Multiplicity	Object snapshots satisfy class-level constraints.
Behavior	Activity outcomes agree with lifecycle transitions.
Architecture	Components and packages support the same responsibilities.

Failure scenario: two concurrent requests

The race

Two librarians request the same available copy.

Both preliminary checks report availability.

Without an atomic rule, two loans could be created.

Required outcome

Only one borrow commits.

The other receives a clear rejection.

No partial loan remains.

The copy has exactly one active loan.

Tests derived from requirements

Scenario	Expected result	Rules
Eligible + available	One active loan and due date	R1, R2
Ineligible member	Rejection, no data changes	R1, R4
Copy already on loan	Rejection, no second active loan	R1, R5
Return active loan	Loan closed, copy available	R3
Concurrent borrow attempts	Exactly one commit	R4, R5

Final capstone and course reflection

Submission portfolio

A short requirements and rules list.

A class model and one object snapshot.

Interaction and lifecycle views.

An architecture view and derived tests.

Review criteria

Correct notation and clear scope.

Consistent names and behavior.

Explicit assumptions and failure paths.

A justified design that you can explain.

Reading and references

Core reading

Dennis, A., Wixom, B. H., & Tegarden, D. (2021).
Systems Analysis and Design: An Object-Oriented Approach
with UML, 6th ed. Wiley. ISBN 9781119559917.

UML notation reference

Object Management Group (2017). Unified Modeling Language,
Version 2.5.1. Structural and behavioral modeling chapters.

Review Lectures 4–11 alongside the integrated library-loan case study.