Software Engineering

Curated computer science study track (not a degree, no licensure claim). Original explanations and practice only, following the reviewed course handout.

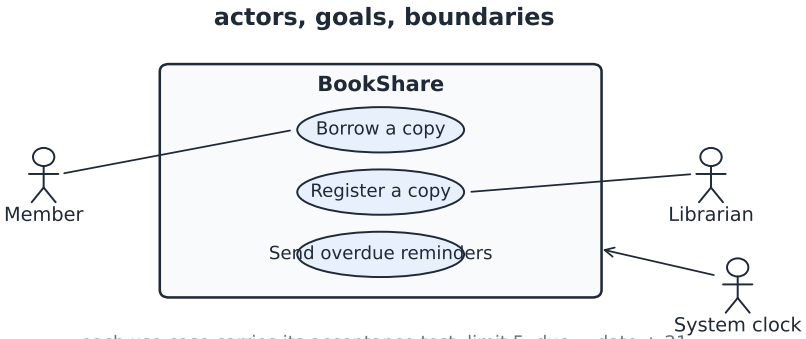
Decisions made visible

- 12.1-12.3 shape the product; 12.4-12.5 hold the line; 12.6-12.7 finish it.
- One running project - BookShare - supplies every artefact, machine-checked in the suite.

Requirements: what and how well

1. **FRs** FR-1 limit 5 (sixth rejected); FR-2 due = date + 21; FR-3 one reminder per overdue day; NFR-1 search p95 under 2 s on 10 000 titles.
2. **Testable** Easy-to-use cannot fail a test; first borrow under 3 minutes measured with 5 first-time users can.
3. **Chain** Every commit that closes a requirement says so; the suite checks the acceptance test exists and passes.

Use-case diagram



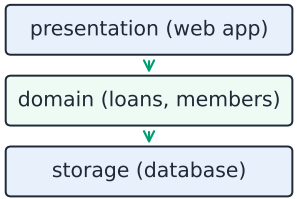
Three use cases: Member borrows, Librarian registers, System reminds.

Architecture: layers pointing inward

1. **Layers** Presentation (web), domain (loans), storage (database); dependencies point inward, never outward.
2. **Counts** Loan-rule change: 1 module, 2 files; smeared across presentation: 3 **layers**, 5 files - the count IS the metric.
3. **Sequence** Presentation calls domain.borrow; domain checks limit and status; storage records with dates - four participants, five labelled messages.

Architecture and sequence diagrams

layers, inward arrows



dependencies point inward

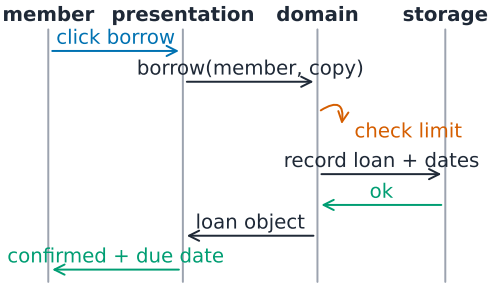
the domain knows no framework

loan-rule change: 1 module, 2 files

smeared version: 3 layers, 5 files

Notifier sits behind an interface

the borrow sequence

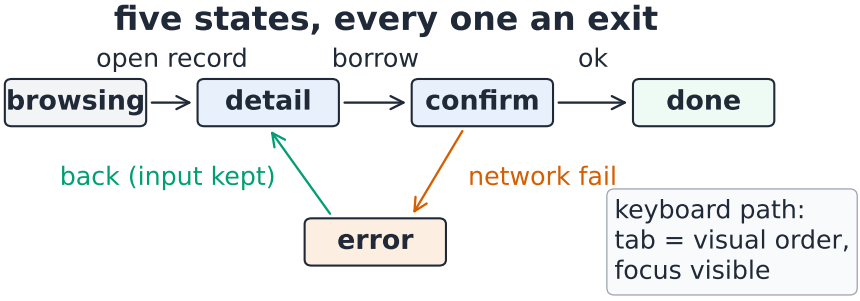


Three **layers** with the inward rule; the borrow sequence labelled.

HCI: states with exits, contrast with numbers

1. **States** Browsing, detail, confirm, done, error - the error state keeps the input and returns to detail.
2. **Contrast** Body text 7:1 passes WCAG AA; the grey disabled at 2.9:1 fails - replaced 4.6:1 with an underline.
3. **Keyboard** Tab order = visual order, focus visible, enter confirms - hover-only menus fail this path.

Interface states



contrast: body 7:1 passes; disabled grey 2.9:1 fails → 4.6:1 + underline

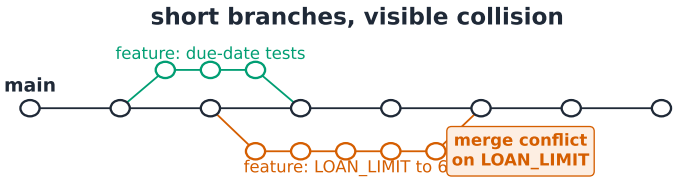
placeholder is not a label; error state keeps the input

Five **states** and **transitions**; the error state always exits.

Version control: the memory that answers why

1. **Model** Short-lived branches, reviewed pull requests; commit dumps kill bisect - 12 runnable commits narrow a regression in at most 4 checkouts.
2. **Conflict** Two branches edit LOAN_LIMIT (6 vs config): the resolution is a decision recorded in the merge message.
3. **Review** Diff + tests + requirement link under ~200 lines - review against the acceptance test, not taste.

Branch graph



one commit = one defensible change, linked to its requirement; 12 commits bisect a regression in at most 4 checkouts

PR contract: diff + tests + requirement link; under ~200 lines

conflict resolution is a decision - record it in the merge message

review against acceptance tests, not taste

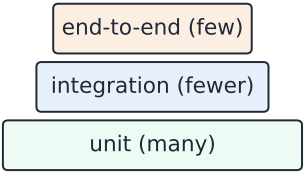
Two features: one merges clean, one collides on LOAN_LIMIT.

Testing: the pyramid and the pipeline

1. **Coverage** 92% line, 81% branch: the gap is decisions never exercised - the branch list is the work queue behind an 85% gate.
2. **Levels** FR-2 at unit level: due = date + 21 with normal, leap February and year-boundary cases; integration reads the loan back; end-to-end scripts the member.
3. **Loop** Push the failing sixth-borrow test: red; implement the limit: green, coverage rises, merge unblocked.

Test coverage and CI flow

the pyramid



coverage: 92% line, 81% branch

FR-2 at all three levels:

unit: +21 days (leap Feb, year end)

integration: the loan reads back

end-to-end: a scripted member

the pipeline (on every push)



red blocks the merge

red-green: a failing test for the 6th borrow

→ CI red → limit check → green → merge

assert behaviour (the due date), not internals

no assertion, no test: just coverage theatre

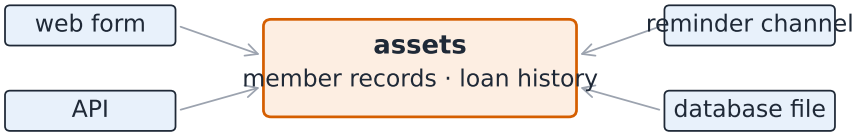
Push triggers install, unit, integration, the 85% gate - green merges.

Security: threats paired with mitigations

1. **Model** Assets (member records, loan history) x entry points (form, API, reminder) x attackers - each threat gets a mitigation and a test.
2. **Matrix** 3 roles x 6 actions = 18 cells, 4 allowed, 14 denials by default; the search form closes injection with parameterised queries (OR 1=1 leaks 0 rows).
3. **Ethics** Selling reading history: documented refusal with alternatives - codes put privacy and public interest above employer preference.

Threat model sketch

assets at the centre, threats paired with mitigations



SQL injection via search form → parameterised queries + length cap
stolen session → data exposure → authorisation matrix: 4 of 18 cells allowed
spoofed reminder channel → signed sender, no loan details in message

least privilege: 3 roles × 6 actions, 4 allowed, 14 denied by default

Assets at the centre; threats paired with mitigations.

Team project: the board tells the truth

1. **Board** Week 4: three stories in flight, the reminder story blocked on the notifier credential - visible, raised, paired on.
2. **Estimate** 8 points/week velocity; 52 remaining = about 6.5 weeks, plus or minus one - recalibrated every **retro**.
3. **Escalate** Blockers older than two days escalate (one in the course history, resolved in a day); the retro changes one thing at a time.

Project board

week 4: three in flight, one blocked

backlog (52 pts)	doing (8 pts)	review	done this week
renew a loan	renew flow (3)	due-date tests (1)	overdue alert (1)
export history	search facet (2)		copy register (3)
multi-branch	**reminders (3) blocked**		

velocity 8 pts/week: 52 left ≈ 6.5 weeks (± 1) · done = CI green + review + docs

blockers older than 2 days escalate; the retro changes one thing at a time

a story without an acceptance test has no done: it has a mood

Week 4: three in flight, one blocked on the notifier config.

Check yourself

- FR-2 says the due date is…? (Borrow date + 21 days.)
- The smeared loan rule touches…? (3 **layers**, 5 files.)
- Bisect over 12 commits: at most…? (4 checkouts.)
- The authorisation matrix allows… of 18? (4.)