Solutions

2.1

- The domain – the limit is a loan rule, not a screen concern ✓.

2.2

- Testability without browser or database, and swap of outer layers ✓.

2.3

- Member, presentation, domain, storage ✓.

3.1

- Violated: the dependency rule (dependencies point inward) and cohesion – the loan rule leaves the loans module ✓.
- Cost: the same rule now touches 3 layers, 5 files instead of 1 module, 2 files – every change multiplies ✓.
- Layered design: presentation renders and collects input; domain decides (limits, statuses, dates); storage persists and reads ✓.
- Moved check: presentation calls storage directly; storage must now validate the limit (rules leak into the database layer) – the sequence gains a presentation-storage arrow and the domain is bypassed ✓.

3.2

- The Notifier interface sits at the domain's edge; delivery implementations live outside – the domain calls the interface only ✓.
- The domain naming email would couple it to a channel that changes; naming nothing keeps the rule pure ✓.
- Swap cost: one configuration line (register push as the implementation); zero domain edits ✓.
- The suite's proof: swap in a test double notifier, run FR-3's acceptance test (reminder count = overdue days) – still green ✓.
- Metric: coupling down (the domain depends on an abstraction, not a channel) ✓.