

# CS-UH 3260: Software Architecture

---

## Project Checkpoint 2 — Quality Attributes & Tactics in a Growing Retail System

### Context — the business evolved

Your 2-tier retail prototype from Checkpoint 1 has grown. The store now runs “Seasonal Flash Sales” and partners with external resellers (VARs/marketplace) who push their inventory and receive orders through your system. Traffic spikes during flash sales; third-party feeds arrive in varied formats; and leadership demands faster releases with fewer outages.

From this point, quality attributes become make-or-break: security, availability, modifiability, performance, integrability, testability, usability. You will specify at least two concrete quality scenarios per attribute (14+ total), map them to tactics/patterns, and implement all the selected tactics .

### Scenario to implement (feature scope)

New Feature Set: “Flash Sale + Partner Integrations”

#### 1) Flash-Sale Orders

- A “Flash Sale” flag on products (start/end time; discounted price).
- During an active window, orders surge. System must throttle fairly, queue work, and keep latency bounded.

#### 2) Partner (VAR) Catalog Ingest

- Ingest a partner’s product feed (CSV/JSON) into your catalog via an intermediary (adapter/gateway or message broker).
- Validate, transform, and upsert items.
- Optionally, schedule periodic ingestion for partners.

#### 3) Order Processing Robustness

- Add retry/rollback around payment/save steps; surface circuit-breaker behavior for flaky external services (mock acceptable).

### Quality attributes: specify scenarios & implement tactics

- 1- For each quality attribute below, **write  $\geq 2$  concrete scenarios using the six-part template (Source, Stimulus, Environment, Artifact, Response, Response-Measure).**
- 2- Map scenarios to tactics/patterns (**at least each quality attribute is mapped to one unique tactic/ pattern, so at least 14 tactics / patterns must be selected**)

3. Implement selected tactics/patterns (**all tactics / patterns mapped in 2 must be implemented**).

**All the following quality attributes must be addressed (two quality scenarios per quality at minimum):**

- Availability (e.g., graceful degradation during flash sale overload)
- Security (e.g., partner feed authentication, protection from malicious input)
- Modifiability (e.g., adding new partner formats without major code change)
- Performance (e.g., bounded latency under 1,000 req/s during flash sales)
- Integrability (e.g., onboarding new reseller APIs with adapters)
- Testability (e.g., automated replay of flash-sale workloads)
- Usability (e.g., clear error feedback for failed orders)

### Approved Tech Stack

Continue with the Checkpoint 1 stack.

Optionally, add Docker/Compose for queues/workers if beneficial.

### Required architectural work

- 1) Quality Scenario Catalog (QS Doc)
- 2) Tactic-level Design with updated diagrams
- 3) ADRs documenting **each** tactic/pattern decision

### Implementation requirements

Implement all tactics / patterns you recommended for each quality scenario. So at minimum you are expected to demonstrate the presence of 14 unique tactics / patterns in your solution.

### Step-by-step tasks (checklist)

1. Ensure repo continuity (from CP1).
2. Draft  $\geq 14$  quality scenarios.
3. Select tactics/patterns for each scenario; write ADR for each selected tactic / pattern.
4. Update all the diagrams from checkpoint 1 to reflect the newly added features.
4. Implement tactics in your system.
5. Add metrics and logging.
6. Test with interface + record/playback.
7. Produce demo video. Within the video, ensure the following is covered:
  - Demonstrate all scenarios of the newly added features.

- Demonstrate how each of the 14 quality scenarios have been satisfied. In particular you have to mention what architectural decision (tactic / pattern) you choose, demonstrate the implementation of the tactic / pattern in your system (by showing the tactic in design / code and also its implication through the user interface) .

### **Deliverables**

- 1) Code repo
- 2) /docs folder with QS-Catalog.md, diagrams, ADRs, Runbook
- 3) Video demo
- 4) Updated README

### **Grading rubric (100 pts)**

- Quality Scenarios (clarity & measurability) ... 28
- Tactic/Pattern selection & justification ... 14
- Implementation of tactics ... 28
- Tests & testability ... 10
- Diagrams & ADRs ... 10
- Demo clarity & repo hygiene ... 10