CS-UH 3260: Software Architecture

Checkpoint 3 — Deployment, Reliability & Returns Module

Due: November 14, 2025

1. Overview

In Checkpoint 1, you built a 2-tier retail application (client + database) with persistence, UML (4 + 1 views), ADRs, and tests. In Checkpoint 2, you extended the system with Flash Sales and Partner Integrations, adding quality scenarios and tactics for performance, availability, security, and testability.

Checkpoint 3 focuses on making your system deployable, observable, and reliable, while also introducing a realistic new feature: Returns & Refunds (RMA workflow).

2. Objectives

- Deployability: run the entire system via Docker Compose.
- Observability: add logs, metrics, and basic monitoring.
- New Feature: implement Returns & Refunds that integrates with your sale flow and payment mock.

3. New Feature — Returns & Refunds Workflow

This feature extends the original Register Sale / Purchase use case by allowing users to return purchased items and receive refunds.

Stage	**Description**	**Responsible Party**
1. RMA Request Submission	Customer submits a return request via portal. Includes order number, reason, and photos.	Customer
2. Validation & Authorization	Support or automated rules validate warranty, purchase date, and eligibility. An RMA number is issued.	Support / System
3. Return Shipping	Customer ships item to warehouse referencing the RMA number. Tracking is logged.	Customer / Logistics
4. Inspection & Diagnosis	Received product is inspected to confirm defect or misuse. Inspection results are logged in the system.	QA / Technician

Stage	Description	Responsible Party
5. Disposition Decision	Decision made: credit, or rejection (e.g., misuse).	QA / Warranty Team
6. Repair / Replacement / Refund	If approved, credit/refund issued; inventory adjusted.	System
7. Closure & Reporting	Case closed; customer notified; records retained for audit; metrics (e.g., RMA rate, cycle time) updated.	System

4. Deployment & Reliability Tasks

A. Containerization

- Create docker-compose.yml for app + database (+ optional queue/worker).
- One-command startup: docker compose up.

B. Observability

- Add structured logging (request ID, timestamp, error level).
- Expose basic metrics (orders/day, error rate, refund/day, etc.).
- Display metrics in a simple dashboard/report to the system admin.

D. Scenario Verification

Pick 2 quality scenarios from Checkpoint 2 (e.g., availability, security, performance) and demonstrate that they are satisfied in runtime through metrics/logs.

5. Deliverables

1. Code Repository with continuity from CP1 & CP2: includes /deploy/docker-compose.yml, Dockerfiles, updated /src with Returns module, /observability/, and updated tests.
2. Documentation (/docs): updated UML (4 + 1 views), ADRs for Docker, observability, resilience, and Returns design, plus Checkpoint3.md summarizing tests, SLOs, and results.
3. Demo Video : show

    - Docker setup (Show that the full system runs correctly using `docker-compose` (all services start successfully).

- Display the monitoring stack (metrics, logs, and traces working). Present the implemented dashboard and clearly explains what every metric is, and demonstrate that it is working.
- return/refund flow: run(simulate) the entire workflow as discussed above and present how the system interacts with every step of the workflow.

6. Grading (100 pts)

Category	Points
Containerization & Setup	20
Logging & Metrics Dashboard (Observability)	30
Returns & Refunds Feature	30
Updated Documentations	10
Demo Video	10