Mining Operations Simulator - Final Report Summary

Group 11: Conor McJannett, Michael Jaimes, Omar Akkad, Niko Tempera

---

Project Overview

The Mining Operations Simulator is a cross-platform desktop application that models procedurally generated underground mining environments. It addresses the dual challenge of maximizing resource extraction while protecting miner safety through an autonomous triage engine that handles both profit prioritization and emergency response. The system serves mining companies as a low-cost planning and risk assessment tool and educational institutions as a training and gaming platform that grades users on safety and efficiency decisions.

Part I: Requirements Description (Sections 1-8)

Scope and Context: The system integrates inputs from geological survey systems, seismic monitoring networks, environmental sensors, and market price feeds. Operations managers and safety coordinators interact bidirectionally. Outputs go to emergency response teams, mining party coordination systems, management dashboards, and educational reporting systems. Explicitly out of scope: legal compliance documentation, financial accounting, hardware equipment control, and physical equipment maintenance scheduling.

Stakeholders: Primary users are operations managers (master-level mining expertise, journeyman tech skills) and safety coordinators (emergency protocol experts needing sub-2-second hazard alerts). Secondary users are training personnel and IT/maintenance staff. Tertiary users are students and gamers who engage with the educational mode.

Constraints: Cross-platform support (Windows 10+, macOS 11+, Ubuntu 20.04+). Zero-budget development using Apache 2.0 / MIT licensed open-source tools. Quad-core processor, 8 GB RAM minimum hardware requirement. Workplace environments include mining operations offices with dual-monitor setups requiring 24/7 availability, and shared computer lab environments requiring save/resume functionality.

Terminology and Data: Core concepts established in this section include Mining Party, Triage System, Procedural Generation, Hazard Alert, and Training Module. Primary data structures defined: MiningParty, GeologicalData, HazardEvent, ResourceNode, TriageDecision, TrackingData, UserProfile, and SimulationSession.

Part II: Requirements Specification (Sections 9-21)

Use Cases (Section 9): Fifteen use cases documented at UC-9-01 through UC-9-15. Four carry full write-up forms: UC-9-01 (Plan Mining Operation), UC-9-02 (Monitor Active Operations), UC-9-03 (Respond to Hazard Alert), UC-9-04 (Review Performance Analytics). The use case diagram shows four actors: Operations Manager, Safety Coordinator, Training Personnel, and Student/Gamer.

Functional Requirements (Section 10): Fifteen functional requirements at F-1 through F-15 cover procedural environment generation, real-time hazard detection, multi-party tracking (60+ simultaneous entities), resource value analysis, triage prioritization, safety response recommendations, training scenario execution, performance grading and analytics, cross-platform compatibility, session persistence, data export, role-based access control, tutorial mode, historical data retention, and administrative configuration.

Non-Functional Requirements (Sections 12-20): - **Look and Feel (Section 12):** Consistent visual hierarchy across dashboard panels; accessibility contrast ratios met (WCAG 2.1 AA). - **Usability (Section 13):** New operations managers reach proficiency within 2 hours of training. Emergency response workflows are completable within 3 clicks from any screen state. - **Performance (Section 14, PR-1 through PR-9):** Hazard alerts generated within 2 seconds of sensor threshold breach. Simulation supports 60+ tracked entities at 30+ FPS. Environment generation completes in under 10 seconds. - **Operational (Section 15, SEC-1 through SEC-5):** Session auto-save at 60-second intervals; full session recovery from last checkpoint after crash. - **Maintainability (Section 16, UH-6+):** Modular plugin architecture for data source parsers; subsystem interfaces documented to support independent replacement. - **Security (Section 17):** Role-based access control segregates operations, safety, training, and student permission tiers. - **Cultural / Legal (Sections 18-20, OE-8):** MSHA safety terminology conventions followed throughout alert language and training scenarios.

Acceptance Test Correspondence (Section 21): Section 21a presents a requirements-to-tests traceability grid mapping all fifteen functional requirements (F-1 through F-15) to corresponding acceptance test cases. Section 21b provides the full acceptance test specifications with preconditions, steps, expected outputs, and pass/fail criteria for each requirement.

Part III: Design Specification (Sections 22-27)

Design Goals and Architecture (Sections 22-24): The system is organized into seven subsystems: DataIngestion, SimulationCore, Triage, Tracking, MonitoringAndAnalytics, TrainingAndEducation, and DataPersistence, with a cross-cutting AccessControl layer. The architecture follows a layered Repository + MVC pattern. The package diagram (Diagram 5) shows subsystem dependencies with strict one-way dependency enforcement: UI layers depend on core services; core services depend on data layers; no circular dependencies.

Design Patterns Applied (Section 25): Four patterns govern the implementation. Singleton controls the single MiningSimulation instance. Observer (via IHazardListener interface and AlertDispatcher) propagates hazard events to registered UI components and logging systems. Strategy (via IDataSource interface) allows runtime swapping of geological, seismic, and market data parsers. Repository pattern (DataPersistence subsystem) abstracts all database operations behind typed interfaces. Template Method (TrainingScenario base class) defines scenario execution flow while allowing subclass-specific scenario content.

System Diagrams (Section 26): Fifteen diagrams produced for the final report, organized as follows: Work Context Diagram (1), Use Case Diagram (2), Domain Model / Class Diagram (3-4), Package Diagram (5), Component and Deployment Diagrams (6, 9-10), Sequence Diagrams for UC-9-01 through UC-9-04 (6-8), Activity Diagram for triage workflow (10), UI Wireframes (11), State Machine for simulation lifecycle (12), and ER/data model diagrams (13-15).

Object Design (Section 27): Detailed object-level design provided for three subsystems. SimulationCore specifies the MiningSimulation singleton, Environment procedural generator, MiningParty aggregate, and TriageEngine. Tracking specifies the EntityTracker, Position/Coordinates3D value objects, and TrackingData record. MonitoringAndAnalytics specifies the AlertDispatcher, HazardEvent, and PerformanceReport with all attributes, types, and method signatures.

Part IV: Project Issues (Sections 28-36)

Open Issues (Section 28): Outstanding questions at submission time concern the specific seismic dataset format to be used for the proof-of-concept integration, the exact scoring weights for the triage priority algorithm, and the final UI framework selection between JavaFX and a web-based rendering layer.

Off-the-Shelf Solutions (Section 29): The system relies on open-source components for procedural noise generation (for environment generation), GIS data parsing libraries, and a charting library for dashboard analytics.

New Problems Introduced (Section 30): Real-time simulation at 60+ entities creates CPU contention with the GUI thread. The chosen mitigation is a dedicated simulation thread with event-queue-based UI updates, decoupling simulation tick rate from render frame rate.

Tasks and Migration (Sections 31-32): Proof-of-concept implementation prioritizes the SimulationCore, Triage, and Tracking subsystems. DataIngestion and TrainingAndEducation are secondary. No legacy system migration required; this is a new system.

Risks (Section 33): Primary risks are scope creep (mitigated by strict requirement freeze after Part II sign-off), performance regression as entity count scales (mitigated by profiling benchmarks at 30, 60, and 100 entities), and data availability for realistic geological inputs (mitigated by USGS public dataset access).

Costs (Section 34): Zero direct cost. Developer time is donated academic effort estimated at 160 combined person-hours across the semester.

User Documentation and Training (Section 35): Planned deliverables include an in-application tutorial mode covering the three primary workflows (plan operation, monitor hazards, respond to alerts) and a PDF reference guide covering all role-specific features. Training Personnel receive an additional facilitator guide for structured classroom use.

Waiting Room (Section 36): Deferred features include comprehensive worldwide seismic modeling, hardware sensor API integration for live data, multiplayer collaborative operations, and mobile client support. These are explicitly out of scope for the Spring 2026 proof-of-concept.

Document Version: Final Report Summary, Parts I-IV
Date: April 2026
Course: CS 440 Software Engineering, University of Illinois Chicago