Course: COSC 2351

Group Members: Oluwateniolafunmi Oguonnoiki, Diana Cesar, Cameren Sudduth
Date: 12/2/2025

Project Description:

Project implements a hospital triage system that registers patients, manages a priority-based triage queue, and logs treatment cases. Patients are stored in a HashMap for fast lookup, queued in a PriorityQueue based on severity and arrival order (FIFO), and treatments are recorded in a chronological ordered LinkedList. The system supports CSV import/export, input validation, and a robust text-based CLI menu for user interaction. A performance demo using deterministic workloads measures enqueue/dequeue operations and allows analysis of algorithmic efficiency. This project demonstrates modular Java design, stable priority queues, and practical use of data structures in a healthcare context.

File Descriptions:

Patient.java – Represents a patient with a unique ID, name, age, severity, arrival timestamp, and arrival sequence for stable triage ordering. Provides getters, setters (where appropriate), and validation.

TriageOrder.java – Implements a comparator for PriorityQueue that orders patients by severity (lower first) and arrival sequence (FIFO for ties)

PatientRegistry.java – Maintains a HashMap<String, Patient> for O(1)-ish lookup by patient ID. Supports patient registration and updating existing records.

TriageQueue.java – Wraps a PriorityQueue<Patient> to manage triage operations. Supports enqueue, dequeue, peek, size, and snapshotOrder (non-destructive listing).

TreatedCase.java – Represents a completed treatment with a patient reference, start/end timestamps, outcome, and treatment notes.

TreatmentLog.java – Stores completed treatments in a LinkedList<TreatedCase> and provides oldest→newest or newest→oldest views.

CsvIO.java – Handles CSV file I/O. Supports loading patient data into the registry and exporting treatment logs with proper escaping.

PerfTimer.java – Utility class for timing blocks of code with try-with-resources to measure performance in milliseconds.

SampleWorkloads.java – Provides deterministic workload generators for testing performance, simulating enqueue and dequeue operations with uniform or skewed severity distributions.

HospitalApp.java – The main CLI application. Handles user input, menu display, patient registration, triage operations, treatment logging, CSV export, and performance demo.