CS 4632: Modeling and Simulation
Project Milestone 1

Eben Nduwimana

09/04/2025

1 Project Proposal

1.1 Project Overview

Project Title: Supermarket Queue Simulation
Domain: Supermarket Checkout System
Problem Statement: The simulation evaluates how the number of queues and cashiers impact wait times, efficiency, and resource allocation. Optimizing the checkout process can result in increased customer satisfaction, reduced workload, and improve business efficiency, which is beneficial in real-world supermarket operations.
Scope:

- Included: Customer arrival, queuing, service time by cashiers, customer departure.
- Excluded: Customer shopping behavior before checkout, employee skill variation, express/self-checkout lanes.

1.2 System Description

System Components:

- Customer: arrivalTime, waitTime, serviceTime, customerID
- Cashier: status (busy or free), serviceTime
- Supermarket Queue Simulator: list of customers, list of cashiers, queue length

System Dynamics: Customers arrive, select the shortest queue, wait for an available cashier, are served, then exit. Cashiers serve customers sequentially. Metrics like average wait time and queue length are collected.

Core Models and Algorithms:

1. Poisson Distribution Algorithm (Customer Arrival)
2. Exponential Distribution Algorithm (Service Times)
3. First-Come, First-Served (FCFS) Queue Discipline
4. Shortest Queue Algorithm
5. Little's Law (Queue Analysis)
6. Cashier Utilization Metric

Assumptions:

- All customers behave equally; no priority groups.
- Customers join the shortest available queue.
- Cashiers work at a consistent pace without interruptions.

1.3 Implementation Approach

- **Programming Language:** Java – suitable for object-oriented modeling of customers, queues, and cashiers.
- **Development Environment:** VS Code or IntelliJ
- **Simulation Type:** Discrete-event simulation
- **Data Collection Plan:** Average waiting times in line, length of each queue, cashier service times

2 Comprehensive Literature Review

2.1 Core Models and Algorithms

1. Poisson Distribution Algorithm (Customer Arrivals) [?]

Description: Events occur independently at a constant average rate. Generates random customers arriving in the supermarket.

Application to Project: Simulates random customer arrivals at the supermarket and provides realistic patterns for queues.

Adaptation for Simulation: In Java, generate the time until the next arrival using a random number generator based on the Poisson process. Each step, generate the next customer arrival time and add the customer to the shortest queue.

2. Exponential Distribution Algorithm (Service Times) [?]

Description: Models how long each customer takes with the cashier to simulate real world, with variation between service times.

Application to Project: Simulates randomness in checkout durations rather than using a fixed duration.

Adaptation for Simulation: Generate service durations using an exponential random variable with a mean equal to the average checkout time. Assign service time to each customer when they reach a cashier.

3. First-Come, First-Served (FCFS) Queue Discipline [?]

Description: Serves customers in the order they arrive (FIFO).

Application to Project: Mirrors real-world supermarket queue behavior; helps track average wait time and queue length.

Adaptation for Simulation: Implement queues in Java using `Queue<Customer>`. Cashier serves first customer in line with `queue.poll()`.

4. Shortest Queue Algorithm [?]

Description: Customer selects the queue with the shortest waiting line to reduce variance in wait times.

Application to Project: Simulates a more efficient supermarket system with multiple queues; measures impact on total wait time.

Adaptation for Simulation: When a customer arrives, Java finds the queue with the minimum size.

5. Little's Law (Queue Analysis) [?]

Description: $L = \lambda \times W$; relates average number of customers in system (L), arrival rate (λ), and time spent in system (W).

Application to Project: Provides a benchmark to verify simulation results and ensure realistic behavior.

Adaptation for Simulation: Track average number of customers and wait times during the simulation.

6. Cashier Utilization Metric [?]

Description: Measures how busy each cashier is to optimize staffing levels.

Application to Project: Ensures cashiers are neither overutilized nor underutilized.

Adaptation for Simulation: Track busy vs. unoccupied time for each cashier in Java.

3 2.3 UML Diagrams

3.1 2.3.1 Class Diagram

Classes:

- Customer: idCounter, id, arrivalTime, waitTime, serviceTime, walkInStore()
- Cashier: isOccupied, currentCustomer, totalTimeOccupied, serveCurrentCustomer(), finishCustomer()
- Supermarket Queue Simulator: idCounter, customerList, cashierList, addCustomer(), removeCustomer(), getQueueLength(), runQueue()

3.2 2.3.2 Activity Diagram

Main Flow: Customer arrives → selects shortest queue → waits for a cashier → served by cashier → exits

3.3 2.3.3 Use Case Diagram

References

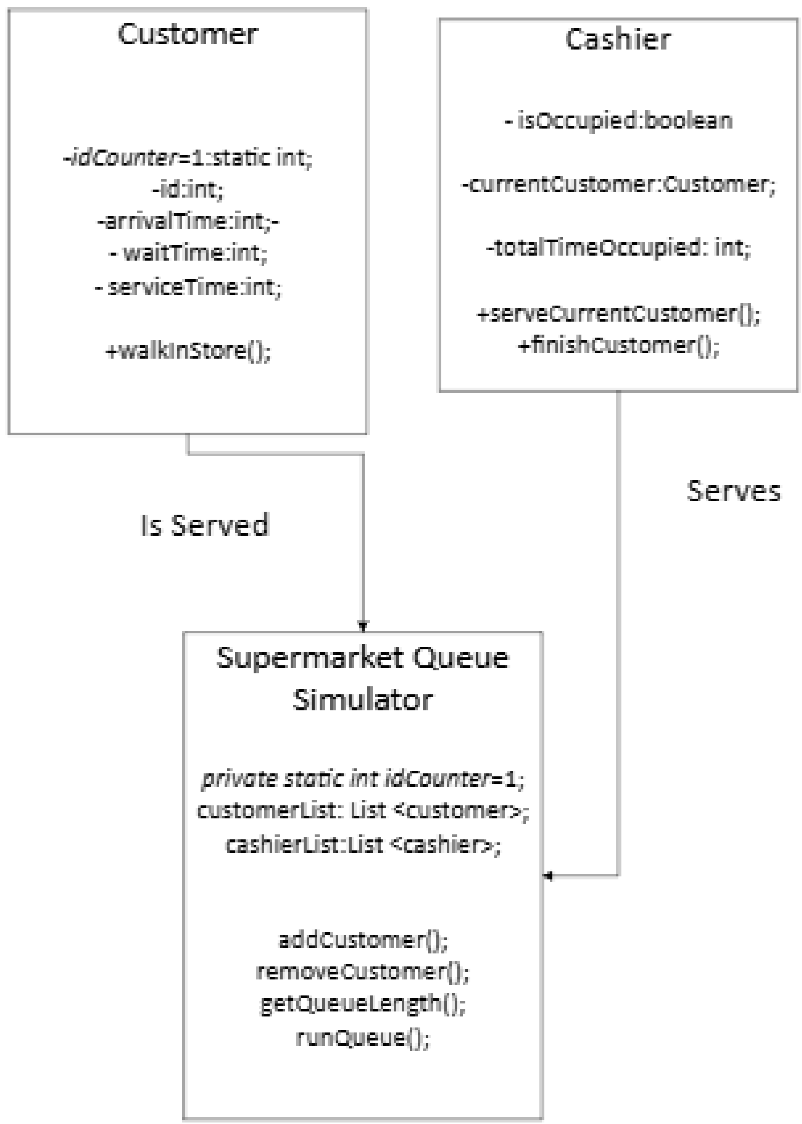

Figure 1: Class Diagram for Supermarket Queue Simulation

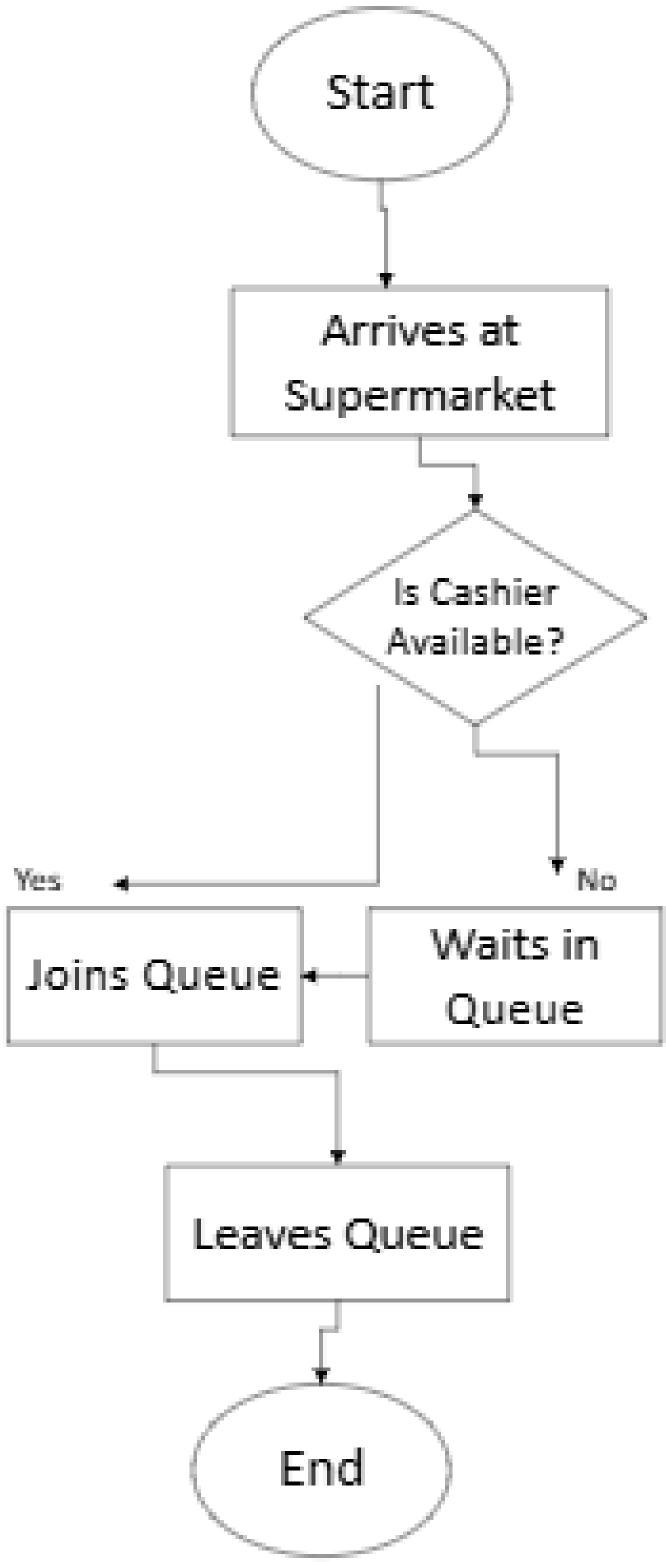

Figure 2: Activity Diagram for Supermarket Queue Simulation

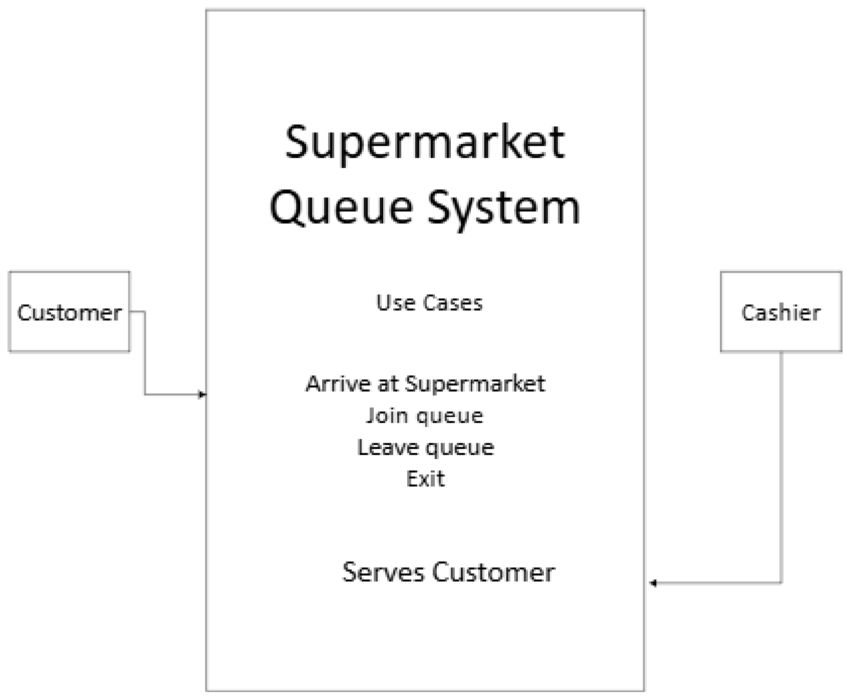

Figure 3: Use Case Diagram for Supermarket Queue Simulation