Supermarket Queue Simulation

Implementation Report

By: Eben Nduwimana

**Introduction**

The following is the implementation report for the Supermarket Queue Simulator, a program designed to model interactions within a real-world supermarket. The simulator focuses on analyzing the behaviors of cashiers, queues, and customers. By running different scenarios, we aim to gather data and generate insights into how to make queues more efficient and how different arrival/service patterns affect customer wait times.

This project is implemented in Java and makes use of object-oriented programming principles such as encapsulation, inheritance, and modular design. The classes developed so far (Customer, Cashier, and CashierQueue) provide the foundation of the simulator. The eventual goal is to incorporate models such as Poisson arrival times and exponential service times, while also allowing different queueing strategies (FCFS vs. shortest queue).

**Implementation Progress**

The following progress has been made in the implementation of the supermarket queue:

* The Main class is the driver of the simulation that creates customers, adds them to queues, and prints out the states of the queues.

**A screen shot of a computer program

AI-generated content may be incorrect.**

* The Customer class has been created with attributes for name and a unique ID.

A screen shot of a computer program

AI-generated content may be incorrect.

* The Cashier class is implemented to represent individual cashiers, each with their own queue and status (occupied/unoccupied).

A computer screen shot of code

AI-generated content may be incorrect.

* The CashierQueue class uses a Java Linked List (implemented as a Queue) to store customers in line.

A screen shot of a computer program

AI-generated content may be incorrect.

The image below displays the simulation in action. Customers can be generated and assigned to a cashier’s queue. Cashier objects can hold their own queues. IDs for both customers and cashiers are automatically increased as new ones are added.

**A screen shot of a computer code

AI-generated content may be incorrect.**

**Design Decisions**

During development, we made several important design decisions to ensure our system remains reliable, usable, and scalable:

1. **Class Design:**
   * Main Class: Drives the simulation by creating customers, assigning them to queues, and printing results.
   * Customer Class: Represents supermarket customers with names and IDs.
   * Cashier Class: Represents workers with their own queue, ID, and status.
   * CashierQueue Class: Handles the underlying structure of each cashier’s line.
2. **Java Queues (LinkedList)**  
   A LinkedList was chosen to represent the queue because it naturally supports the FIFO (First-In-First-Out) behavior, aligning with how real-world queues operate. In addition, Linked Lists have efficient insertion and deletion operations, perfect for adding and removing customers as they are served.

A screen shot of a computer code

AI-generated content may be incorrect.

The design of the simulation makes it easy to integrate more advanced features later in development, including shortest queue selection (choosing the queue with the fewest number of customers) and varying waiting times for each queue.

**Initial Testing**

To ensure quality in the second milestone of the supermarket queue simulation, the following tests were conducted:

* Customers were added to multiple queues, and the simulator correctly assigned them to different cashiers.
* Unique IDs (customerID and cashierID) were successfully generated for both customers and cashiers.
* Console outputs matched expected behavior as customers entering and leaving queues.

During the next series of tests, we will focus on improving the realism of the simulation when compared to real world supermarket flow. For example, a Poisson arrival process will be implemented, generating customers at more random intervals to better reflect real world supermarket behavior. In addition, we will be including service times to track how long each customer spends in line. We will then analyze the data to discuss how we can improve efficiency in the future.

**Conclusion**

The project is on track with the initial proposal. The primary classes have been successfully implemented, initial testing has shown that features are implemented correctly, and the system design supports scalability in the future. During the next milestones, we’ll focus on the project as we move closer to providing valuable insights into real-world queue management.

The repository for this project can be found at: https://github.com/xscape24/Supermarket-Queue