# Grocery Store Simulation Project

**Abstract**

In this project, I simulated three different strategies for managing a grocery store checkout line to study their efficiencies. The three strategies included (S1) separate lines for each register, (S2) a single line for all registers, and (S3) two shared lines. I implemented the simulation in Java using ArrayLists, which allowed dynamic resizing of queues as customers arrive. My results showed that while the single-line strategy minimized maximum line size, it produced the highest average wait time. By contrast, individual lines produced the lowest wait times but created very long queues. These findings show trade-offs between fairness,  space usage, and efficiency in checkout queue system design.

**Results**

- **Experiment Design**

I wrote Java code to simulate three queue management strategies: S1 (separate lines per register), S2 (single shared line), and S3 (two shared lines). Customers were randomly assigned to registers depending on the strategy. For each strategy, the simulation tracked two metrics: average wait time (in minutes), which is how long a customer spends waiting before checkout; and maximum line size, which is the longest observed queue length across registers.

I ran the simulations with consistent customer arrival rates and service times, so differences would reflect only the strategies.

- **Hypothesis**

Before running the experiment, I hypothesized that S1 (single-line strategy) would produce the fairest distribution of waiting times but might increase the overall average wait time, since everyone must wait behind the entire line. I also expected S1 to produce the fastest average service for lucky customers, but at the cost of very long individual lines.

- **Data**

The simulation results are listed below:

| Strategy | Average Wait Time (minutes) | Maximum Line Size |
|---|---|---|
| S1 (separate lines) | 2.1 | 20 |
| S2 (single line) | 5.9 | 7 |
| S3 (two lines) | 2.5 | 10 |

**Figure 1: Comparison of Average Wait Times**: The bar chart shows that S1 had the shortest average wait time (2.1 minutes), while S2 had the longest (5.9 minutes). This means that although the single-line system reduces congestion, it results in slower customer throughput overall.

**Figure 2: Maximum Line Size Across Strategies**: The maximum line size was smallest for S2 at 7 people, compared to 20 for S1 and 10 for S3. This implies that the single-line strategy uses store space more effectively, preventing long queues from blocking aisles.

- **Summary**

The results show a clear trade-off between fairness, efficiency, and space usage in checkout line management. S1 is best for minimizing customer wait time on average, but it creates extremely long lines at individual registers. S2 reduces congestion and ensures fairness since all customers wait in a single queue, but at the cost of longer waiting times. S3 provides a balance between the two. These results indicate that the design of the checkout queue system depends on whether wait time or preventing congestion is the higher priority.

**Acknowledgements**

I would like to thank the course staff for providing the project template and guidance. I consulted the W3Schools Java Tutorial for review.