Kyle Kaminski, Matt Keville, Victor Caceres, Tim Gitt

Professor Safko

Introduction to System Simulation and Modeling

May 4th, 2020

Final Project Report

For our final project for Introduction to System Simulation and Modeling, we decided to create a Queueing system with multiple customers and multiple servers (think like a grocery store). The system contains Customer objects and Server objects. There can be as many customers created as needed, but there are only two servers from the start. There cannot be any more servers added throughout the program, but one of them can be opened and closed as needed. We have also implemented a global time, which increments every time a customer is created, or when a customer is being processed. This global time is used for processing customers and deciding when they should balk, renege, or jockey (will be explained later in this document).

The servers are remarkably similar but have some differences to distinguish them. The servers both accept customers and process them as time goes on (user defines when time moves on). They both accept as many customers as they want. They both have the open and close functionality, although we only ever open and close the second server. The servers both implement a queue system with the FIFO implementation of add/remove, meaning that the first customer that got in line is the first customer that will be processed, and the last customer added is always in the back of the line.

Although the servers use the same object class, they do have some unique difference within our program. The first server qualifies as our default server that will always remain open. If the other server is closed, customers will always default to this server when they want to be processed. This server has no limit on the cart size; therefore, no customer will be excluded from entering.

The second server has special attributes that differentiate it from the first server. First, it is not always open, and can be manually opened and closed by the system. When it is closed, it will never take customers. When it is opened, system has an algorithm to decide which customers goes in this server, and which customers go in the other server. This is based on the size of their carts. When the server is closed, customers can once again no longer be added to the server; if the server had customers in it at the time of being closed, they will automatically be added to the back of the default server (consolidating the lines into one). Lastly, customers can only be added to this secondary server when they have a specified cart size of less than or equal to 5 (and this server is open, of course). Therefore, this server can be seen like an Express Line in a grocery store.

The system also implements a way to process the customers. When menu option 2 is selected, the first customer in each respective server is partially processed. This decrements their processing time attribute. When their remaining amount of processing time is 1, then the next time the time is incremented, they are fully processed; they are then removed from the server, and the new first customer was the old second customer in the line. The line then gets shorter by a value of 1, since 1 customer was processed and removed.

The system can manually open and close the second server only. These are both done with menu option 3 and menu option 4, respectively. Like mentioned above, when the second server is open, customers with the specified cart size can be added to it. When the second server is closed, it can no longer take new customers. Customer that were waiting in this line are then added to the default server’s line.

The system can report on q\_hat, u\_hat, and B(t). In this course, q\_hat is used to define the average number of customers in the server at any given time. When menu option 5 is selected, this is reported to the user. Next, the system can report on u\_hat, which is used to define the average utilization time of the server over the course of a run; in other words; this is reported using menu option 6. Finally, menu option 7 reports on B(t), which reports whether a server is busy at any given time. This simply means we return 1 if there is at least 1 customer in the line, and 0 if the server is empty.

The system can report on balking, reneging, and jockeying. Balking is when a customer can decide not to join the server in the first place. When this happens, the customer will never be added to a server and will be garbage collected. Reneging is when a customer wants to leave the server prematurely and does not get served. When this happens, the customer will be removed from the server, and the customer behind them will take their spot in line. Lastly, jockeying is when the customer wants to switch lines. When this happens, they are taken out of their spot in line (the customers behind them move up), and they are added to the back of the other line, if applicable.

Finally, the system can be terminated when the user is done. Option 11 terminates the program for the user.