Systems Simulation and Modeling  
Professor Gregory Safko

Rowan University

**Final Team Group Project**

**(Modified: Q-2020)**

Due: Presentations start on April 27, 2020

Final Reports, Deliverables and Presentations by: May 4, 2020

**Design and Analysis and a System Model (Modified)**

**Project Basics and Expectations:**

For this team project, you will write a program that will calculate the wait, service, and busy time of some server system for some queuing system of your choice.

Components:

You will have a generic customer (a person, a product, a process, an assembly), with some unique ID, that will track their entry, wait, and process times from a queue to a server

You will have the ability to open and close a second server (but the first will never close). If a second server opens, a secondary queue must also open. Customers can also track if they want to perform one of three state options: balk, renege, or jockey.

A recap of the definitions are as follows:

Balk: A customer can decide to not join a queue in the first place

Renege: A customer can decide to leave the queue without being served

Jockey: A customer can decide to jump to another server if that server opens.

You can model:

~~Assembly Line / Processing Systems~~

~~Food Industry Systems (includes food processing and food services)~~

~~Traffic Movement and Pattern Systems~~

~~Storage/Inventory Systems (with or without stock rotation)~~

**Single or Multiple Queued systems**

Your code will have elements of balking, reneging, and jockeying

Welcome to the SSM Queue and Server System

Team: Hi I Queue

Members: AlbertE, ThomasE, NikolaT, IsaacN

Select an Option for the System:

1. Create a [Customer/Item]

2. Process a [Customer/Item]

3. Open the second server

4. Close the second server

5. Find q-hat

6. Find u-hat

7. Find B(t)

8. Report on balking [customers/items]

9. Report on reneging [customers/items]

10. Report on jockeying [customers/items]

11. Quit

Some notes:

You may need to design some sort of look-ahead in your queue (or perhaps a glass queue). Therefore you may find it necessary to write your own queue objects.

All times are generate by your own choice of a PRNG (see below). Design your PRNG to throw in some outliers that make the queue behave differently based on the numbers (such as a large wait or process time)

Menu choice 1: Create a customer.

Create a customer (give them a unique name if they are people, a unique part number if they are a product, or some incremental id if they are a stock item). You made need to track some internal numbers, such as quantity, for example a customer with less than 5 items can go to a “5 items or less queue/server”, or a part with more than 6 components might go to a server designed to handle a more complex [6 or more components] part.

After the customer is created, they automatically go into the queue.

Menu choice 2: Process a customer.

The customer at the front of the queue is served, and the wait times and process times are recorded.

Also, if two queues are open, ask if the process is addressing the customer in the primary queue or the secondary queue.

Menu choice 3: Open the second server

This will only open a second server (and the secondary queue) if there is no operating second server.

Menu choice 4: Close the second server

You should only close the second server if the secondary queue is empty, but there are possibilities that you can close the second server. (e.g. server failure, or a huge process time is encountered) (Do you want to put these customers in the back of the primary queue?) Under what circumstances is this viable?

Some guidelines for Balking:

You can decide your own level of balking. You might say this if queue.getSize( ) > 5, or if the sum of the wait time then a customer balks. Delete (balk) the customer, and track the event.

Some guidelines for Reneging:

You can decide your own flag for reneging. Does it happen statically or dynamically? What triggers a renege? Process (delete) the reneging customer, and track the event.

Some guidelines for Jockeying:

A customer can only jockey if a second server is open. Only the first person in line can Jockey to an empty server queue (for Server 2). But do others jockey? What is your criteria? Does it come from the front of the primary queue? Can someone in the middle of the primary queue jockey to the secondary queue? Can new customers go right to the secondary queue? Should the secondary queue grow larger than the primary queue? The choice is yours. Process the jockeying customer, and track the event.

Generate your random numbers using one of the methods we discussed in class (Uniform, Poisson, Weibull, Beta, or Gamma). Your choice. Do not use a built-in PRNG tool.

**Deliverable Rubric:**

The team leader will submit a single report on your study. The report should be a minimum of three pages per teammate in length, and should include all necessary graphics and data tables. However, exclude the amount of page size taken up by graphics and data tables in your final page count. Margins should be no more than 1 inch, and font should be no more than 12 pitch. You may attach any other artifacts as necessary (namely, data collection in Excel, PowerPoints, Prezi links, Animations)

**Project Grading:**

Your final grade for the project is calculated as follows:

The Final Deliverable: 60% (The final reports, powerpoints, code, etc.)