Embedded System

Project 4 Code Approval

*Date of Submission: March 26, 2019*

Joel Yuhas : [jxy8307@rit.edu](mailto:jxy8307@rit.edu)

Athaxes Alexandre : [axa2012@rit.edu](mailto:axa2012@rit.edu)

**Analysis/Design**

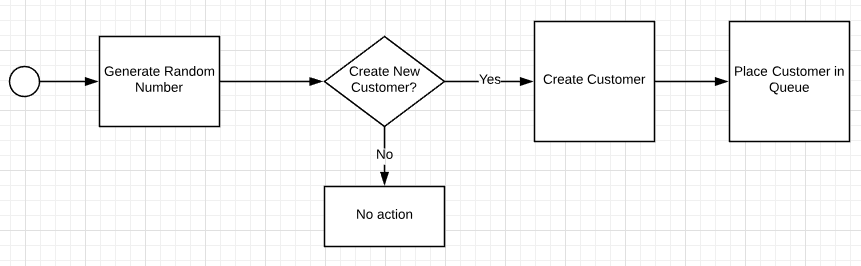
After reviewing the requirements of the project, the following layout has been chosen to implement the program. In total there will be two different classes of threads and four different data objects. The threads will be used to modify the objects and “move” the customers around the bank in specified times. The objects are designed to store the appropriate information. These objects will most likely be structs. The names and descriptions are below.

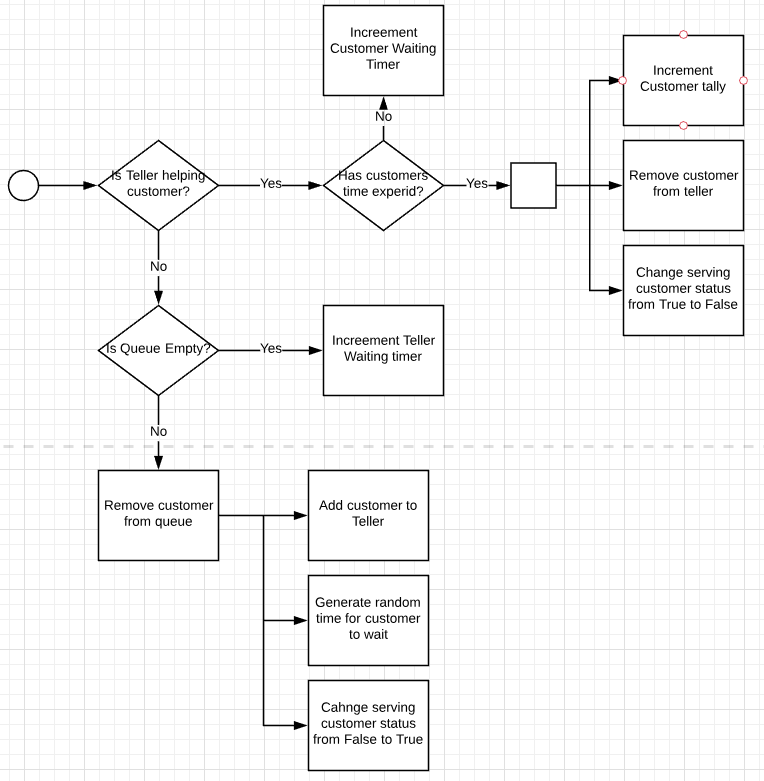
**Threads**

* Bank Thread
  + Responsible for initializing new customers and adding them to the queue in random intervals. Also responsible for enforcing start and end time.
* Teller Thread
  + Responsible for removing customers from the queue, keeping them at a teller for a random amount of time (in a certain interval), and eventually removing them from the teller and the bank. Also continually checks if the teller has a customer at the station. If not, then add one from the queue. This thread can be replicated to add more tellers. The benefit to this design is that any number of tellers can be added without needing to modify other items.

**Objects**

* Customer Object (struct)
  + Customer ID (int)
  + Time spent in Queue (float)
  + Time spent at teller (float)
* Teller Object (struct)
  + Is currently serving customer (Boolean)
  + Total # of customers served (int)
  + Time spent waiting for customers (total) (float)
  + Customer (customer struct)
* Queue Object (struct)
  + Number of customers in queue (int)
  + List of customers (list of customer structs)
* Bank Object (struct)
  + Total # of customers (int)
  + The current time (float)



**Figure 1: Bank Thread**

**Figure 2: Teller Thread**