

# Airline Check-in System – Design Document

CSC360 Fall 2018

Zhe(Kevin) Chen

**1. How many threads are you going to use? Specify the task intend each thread to perform.**

There will be thread for each customer and each thread(customer) will play the role as customer. There will also be thread for each clerk and each thread(clerk) will play the role of clerk. Their specific job will be described in question 8.

**2. Do the threads work independently? Or is there an overall controller thread?**

Customer and clerk threads are working independently since each customer is independent. There will be no controller thread other than main thread that creates customer and clerk thread.

**3. How many mutexes are you going to use?**

There will be two mutexes for each queue, four mutexes for each clerk, two mutexes for the calculation of average time for each business class.

**4. Will the main thread be idle?**

Yes, the main thread will be idle after created all the threads needed, it waits for all the customers to be served(exit).

**5. How are you going to represent customers? Data structure.**

There will be several data in the Customer data structure: customer id/class type/arrival time. Customer queue(both business and economic class) that holds customers will be implemented using FIFO linked list.

**6. How are you going to ensure that data structures in your program will not be modified concurrently?**

Pthread mutex is the solution to make sure no such thing will happen in our design, it works in a way that only one process can enter its critical section such as read and write to queues.

**7. How many convars are you going to use? a) Describe the condition that the convar will represent. b) Which mutex is associated with the convar will represent. c) What operation should be performed once pthread\_cond\_wait() has been unblocked and re-acquired the mutex.**

I will use four conditional variables for clerk and two for queue.

a) Clerk conditional variables are used when each clerk finished signaling queue will wait for customer to finish check-in. Customer conditional variables are used when customer arrived this customer will have to wait on these conditional variable and wait for clerk to send signal.

b) The queue mutexes are associated with clerk conditional variables. This is because when the customer being called it will be leaving the queue which requires read and write to the queue. The clerk mutexes are associated with clerk conditional variables.

c) When `pthread_cond_wait()` has been unblocked and re-acquired the mutex, in situation where customer is waiting in a queue, it will lock the queue and when it finds it is the head of the queue, it will leave the queue and start check-in. And in the situation where clerk is waiting for customer to finish check-in it will wake up and try to serve next customer.

**8. Briefly sketch the overall algorithm you will use.**

**1) Main thread**

set up the initial values and attributes  
read file and create customer thread for each customer which contains their information,  
create all customer threads  
create four clerk threads  
while there is still unfinished customer thread do nothing  
calculate and print all the stats

**2) Customer threads**

(assume it already has all the information on this specific customer)  
sleep for its arrival time  
mutex(based on class type) lock  
when the queue is busy, unlock the queue and re-acquire the lock  
enter a queue based on its class type  
wait for the signal  
check if this customer is actually the one being called  
if no go back to wait (done in while loop)  
if yes customer will leave the queue  
mutex unlocked  
report current time as service start time  
update stats overall queue waiting time  
get served(sleep for its service time)  
report current time as service finish time  
signal clerk that service is finished

**3) Clerk threads**

loop  
queue mutex lock  
if any queue is busy, unlock the queue and re-acquire the lock  
check if any queue has customer(business class has the priority)  
if yes  
signal the selected queue  
queue mutex unlock  
clerk mutex lock  
wait for customer to finish check-in  
clerk mutex unlock  
back to top