CIS*2460 Assignment 3

Question 1

(a) See appended design section.

(b) The following are the outputs of running my program for 1, 5, 100, and 1000 days.

```
overkill@overkill:~/code/CIS2460/A3$ ./Q1 1
Running for 1 days!

>> average teller queue size (= average wait in minutes): 0.000000
>> average cashier queue size (= average wait in minutes): 0.168750
>> Cashier queues were bigger than teller queues.
>> probability of getting an idle teller: 1.000000
>> probability of getting an idle cashier: 0.443750
>> Total customers 36
overkill@overkill:~/code/CIS2460/A3$ ./Q1 5
Running for 5 days!

>> average teller queue size (= average wait in minutes): 0.000000
>> average cashier queue size (= average wait in minutes): 0.164583
>> Cashier queues were bigger than teller queues.
>> probability of getting an idle teller: 0.990000
>> probability of getting an idle cashier: 0.553333
>> Total customers 190
overkill@overkill:~/code/CIS2460/A3$ ./Q1 100
Running for 100 days!

>> average teller queue size (= average wait in minutes): 0.000812
>> average cashier queue size (= average wait in minutes): 0.152708
>> Cashier queues were bigger than teller queues.
>> probability of getting an idle teller: 0.992750
>> probability of getting an idle cashier: 0.561354
>> Total customers 3818
overkill@overkill:~/code/CIS2460/A3$ ./Q1 1000
Running for 1000 days!

>> average teller queue size (= average wait in minutes): 0.000877
>> average cashier queue size (= average wait in minutes): 0.167212
>> Cashier queues were bigger than teller queues.
>> probability of getting an idle teller: 0.991652
>> probability of getting an idle cashier: 0.563875
>> Total customers 38579
```

For further investigation, feel free to compile and run the programs for yourself. The instructions to do so are in the readme.txt file.

Question 2

(a) See appended design section.

(b) I ran my program for 1000 days:

```
overkill@overkill:~/code/CIS2460/A3$ ./Q2 1000
Running for 1000 days!

>> average wait for customers who got help on the floor: 0.014123
>> average wait for customers who just wanted to be cashed out: 0.010321
>> average wait for both types of customers: 0.024444
>> probability of an idle teller: 0.871725
>> Total customers 38421
```

i. Comparing the results, it becomes clear that the overall wait time in the redesigned system is much more effective.
ii. The average wait time for a customer who wants to buy an expensive item is greater in the redesgned simulation. However this is just a tradeoff of getting rid of the customer queue.
iii. The average wait time for those who want to buy items off the floor is strongly reduced in the redesign. This is a result of having three tellers to wait for instead of a single cashier.
iv. The average idle time for a teller has decreased, which means they are being more effectively utilized. In the first design, they were very underutilized, meaning a drop in productivity. This redesign increased productivity by having them idle less. Yet, the idle time is still pretty high. The simulation could probably afford to remove one more teller without having overwhelming demand.

Question 3

(a) See (b)

(b)

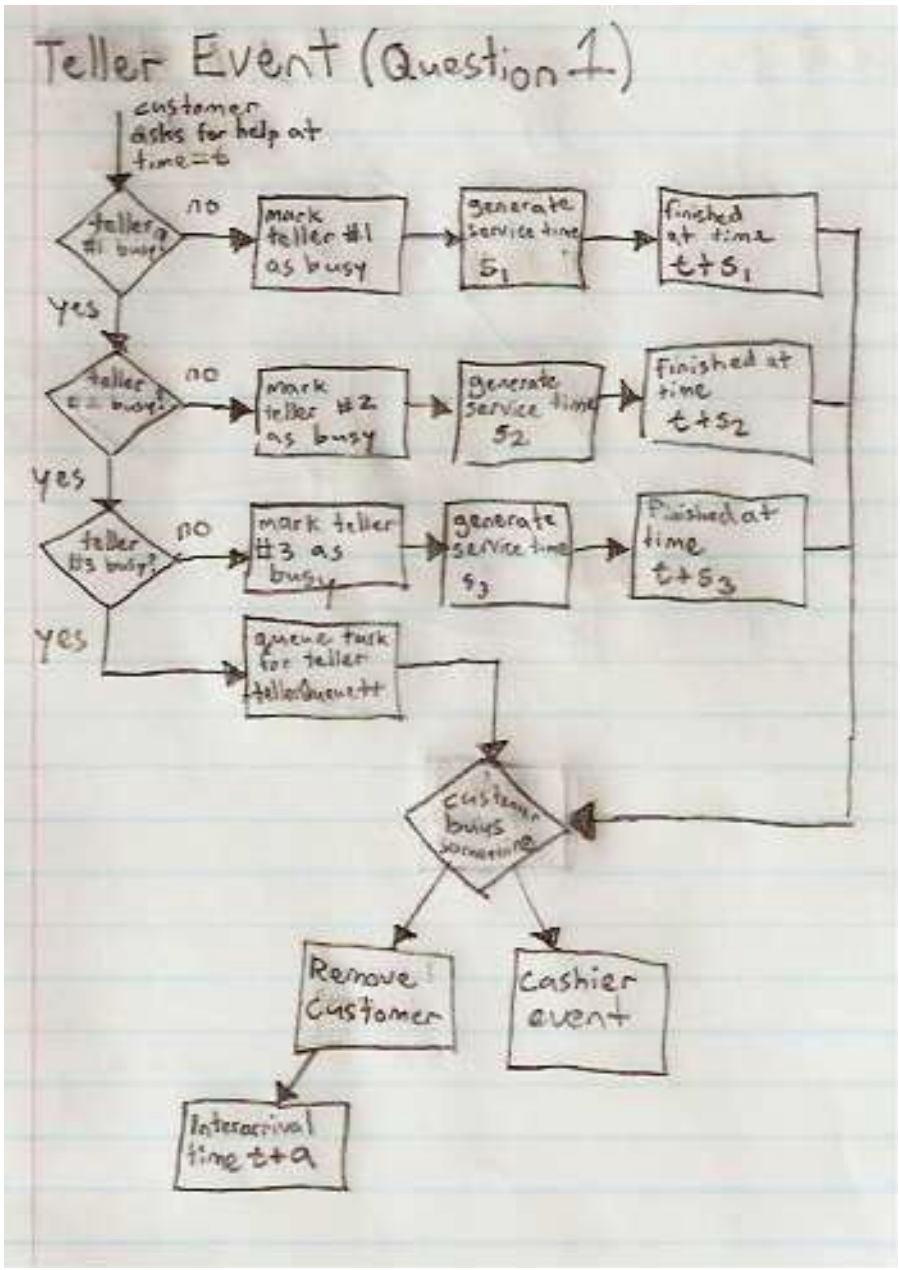
```
overkill@overkill:~/code/CIS2460/A3$ ./Q3 90
>> Running for 90 days... (may take a bit to run events)

>> min time = 0.253000
>> max time = 3.233700
>> average time = 0.861872
>> berth 1 usage: 0.539472
>> crane 1 usage: 0.580500
>> berth 2 usage: 0.324834
>> crane 2 usage: 0.580460
>> total ships 107
```

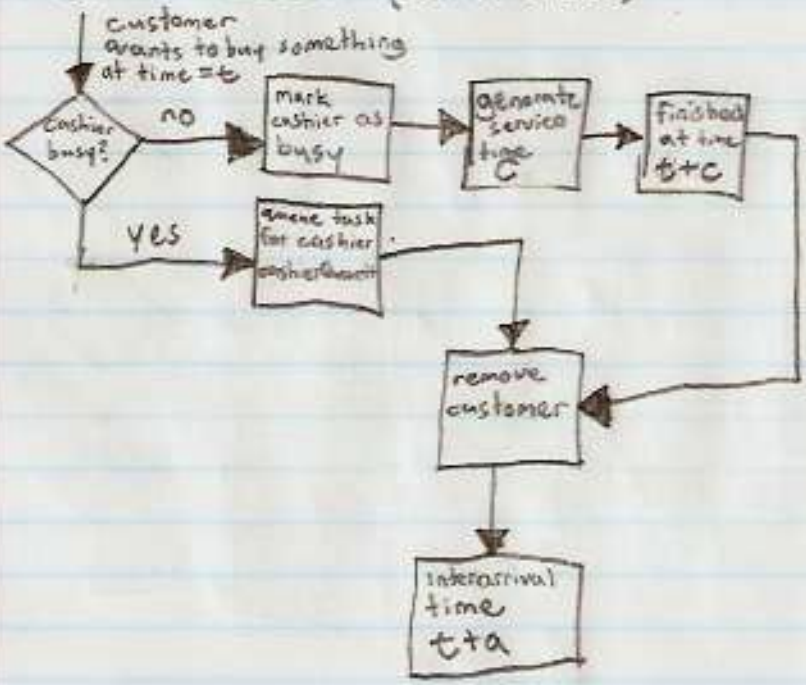
The “expected” usages are calculated from the average of usage over all days of the simulation. See appended design section for design details of this simulation and others.

Design Information

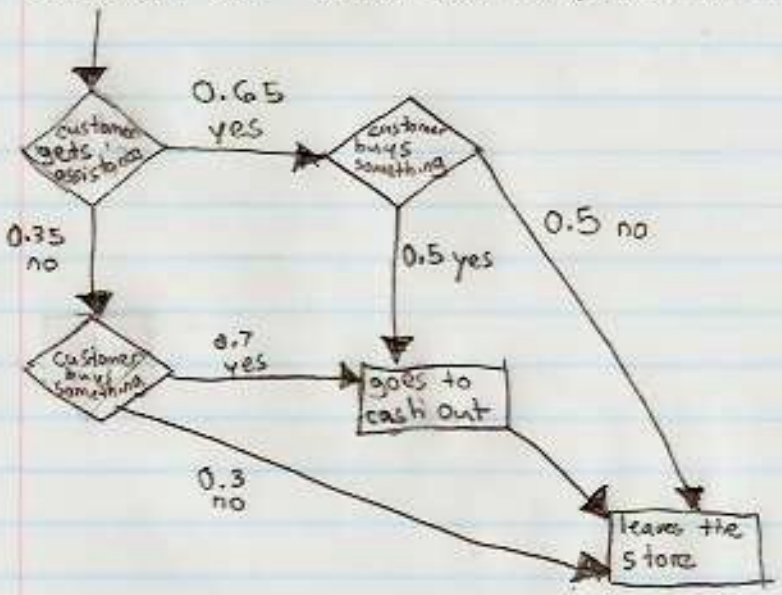
Below are the designs I used to implement my simulations. Sorry for the poor scan quality, but I think the text should be legible provided you zoom in a bit with your PDF viewer. Of course, a good way to check implementation details is also the source code.


Teller Event (Question 1)
customer asks for help at time=t
teller #1 busy?
no
mark teller #1 as busy
generate service time s_1
finished at time $t+s_1$
yes
teller #2 busy?
no
mark teller #2 as busy
generate service time s_2
finished at time $t+s_2$
yes
teller #3 busy?
no
mark teller #3 as busy
generate service time s_3
finished at time $t+s_3$
yes
queue task for teller
customer buys something
Remove Customer
cashier event
Interarrival time t+a

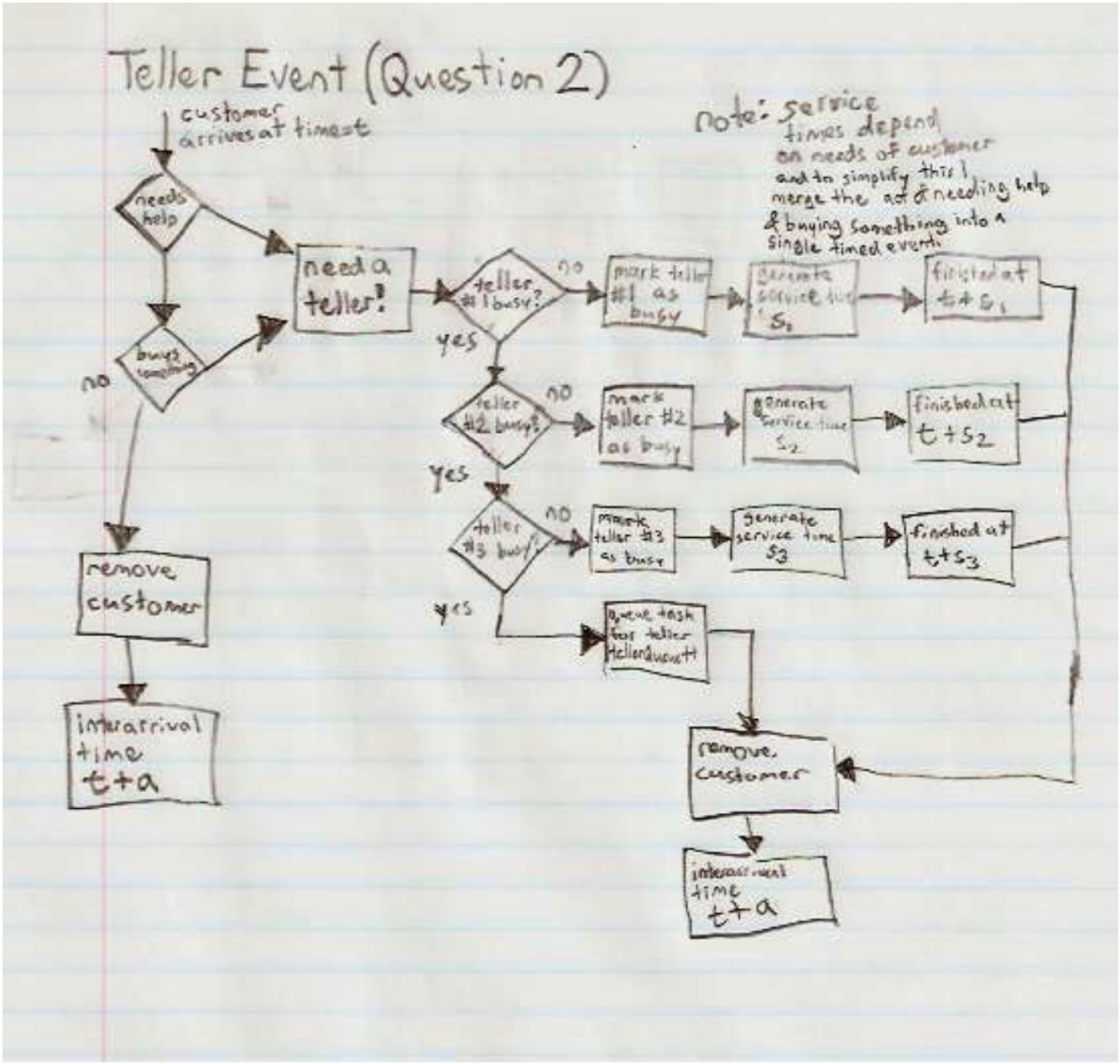
Cashier Event (Question 1.)

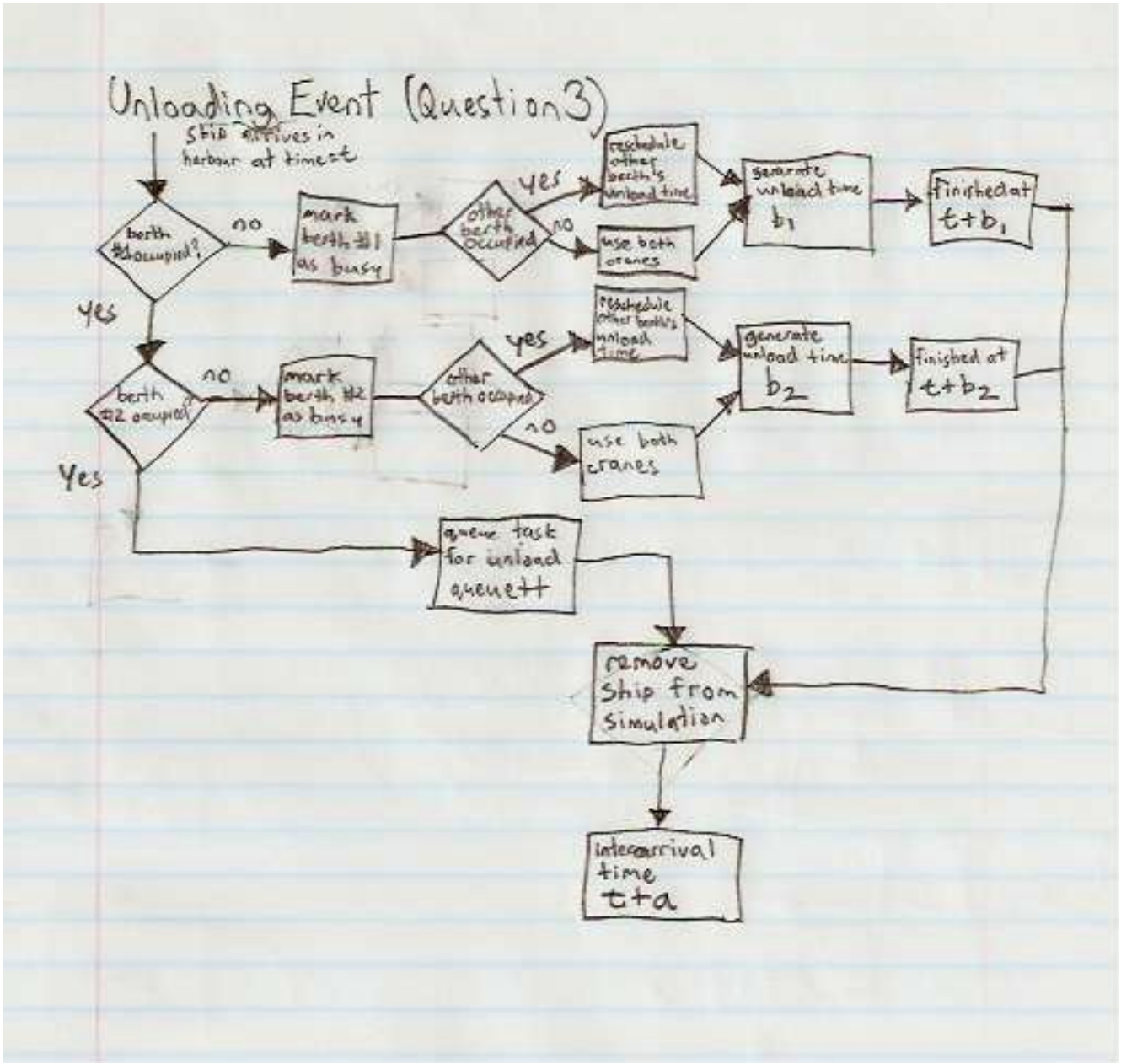


Customer Decisions (Question 1 & Question 2)



Teller Event (Question 2)




Unloading Event (Question 3)
ship arrives in harbour at time=t
berth #1 occupied?
no
mark berth #1 as busy
other berth occupied
yes
reschedule other berth's unload time
no
use both cranes
separate unload time b_1
finished at $t+b_1$
yes
berth #2 occupied?
no
mark berth #2 as busy
other berth occupied
yes
reschedule other berth's unload time
no
use both cranes
generate unload time b_2
finished at $t+b_2$
Yes
queue task for unload queue++
remove ship from simulation
interarrival time t+a