

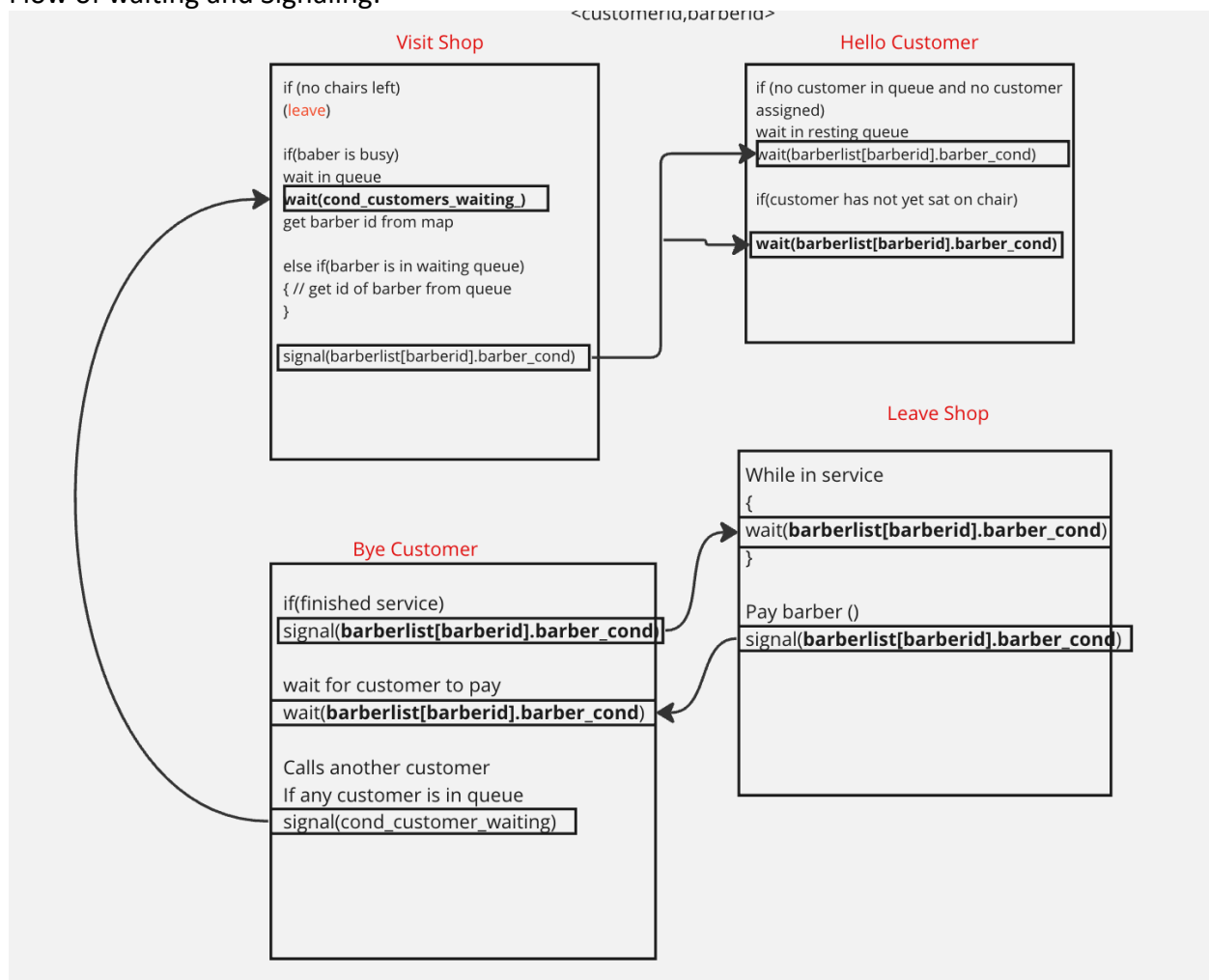
Report for Program 2

Documentation:

Data Structures used: 2 Queues, 1 Map, 1 Vector:

1. This program contains a struct Barber which contains information of the barber including his id, current customer, currently servicing state, 2 condition variable **barber_cond** , and **finished_signal**.
2. The vector barber_list contains all the Barber objects initialized in init() method.
3. resting_barber queue keeps track of sleeping barbers, and waiting chairs keep track of waiting cutomers.
4. Once when a Barber is assigned to a customer, it adds the value to the Map which has a key of Customer and value as Baber assigned.

Flow of waiting and Signaling:



- If barbers are in waiting queue then customer signals the barber waiting in Hello customer method. If all barbers are busy then the customer waits for the barber to signal which it does at the end of bye customer method.
- The customer moves to the service chair and signals the barber which is waiting in hello customer.
- The customer waits for the barber to finish service.
- The barber signals the customer after finishing service and waits for the customer to pay
- The customer after getting signal of finish pays the barber and signals the barber
- The barber checks if any customer waits in the queue and signal them using a common condition. Rest all the condition are specific to each barber.

Keeping track of customer Id for barber and barber Id for Customer:

- The program uses currently_in_service map and barber array to achieve this
- If customer pops the barber from the waiting queue it adds an entry to the currently_in_service ;
- If the barber pops the customer waiting in queue it adds an entry to the currently in service queue.
- After the service the barber removes the entry from the map.
- Barber also has its current customer stored in its struct

Discussion:

Step 5: Run your program with

```
./sleepingBarbers 1 chair 200 1000
where chairs should be 1 ~ 60.
```

Approximately how many waiting chairs would be necessary for all 200 customers to be served by 1 barber? *Note that depending on the number of cores and clock skews, it may very well require over 60 chairs to accomplish the task.*

It depends on the time the customer visits. If all the customer visit at same time then 120 would be sufficient.

Minimum required chairs would be 92 – but it is not guaranteed that 200 customers will be served

Guaranteed required chairs will be 100

Step 6: Run your program with

```
./sleepingBarbers barbers 0 200 1000
```

Where *barbers* should be 1 ~ 3.

Approximately how many barbers would be necessary for all 200 customers to be served without waiting?

86 barber would be necessary or minimum barbers to serve all the customers. But it still depends on when the customer comes to the shop.