Project Two Design
John Lawler | JML190001 | CS4348.501

SEMAPHORE LIST

Semaphore Name: Description. Initial value: x.

- **frontdesk_ready**: indicates whether the front desk is ready to handle a customer. Initial value: 0.
- **customer_exchanged**: indicates that a customer has completed their interaction with a hotel employee (either bellhop or frontdesk). This has an initial value of 1 so that the first customer is not waiting and can continue immediately. Initial value: 1.
- **checked_in**: indicates that a customer has successfully checked in at the front desk. Initial value: 1.
- **give_room[]**: an array of semaphores where each semaphore represents the event of the front desk giving a customer a room. Initial value: 0.
- **bellhop_ready**: indicates whether a bellhop is ready to assist a customer. Initial value: 0.
- **get_bags[]**: an array of semaphores where each semaphore represents the event of a bellhop acquiring a customer's bags. Initial value: 0.
- **enters_room[]**: an array of semaphores where each semaphore represents whether the customer has entered their room. Initial value: 0.
- **give_bags[]**: an array of semaphores where each semaphore represents the event where a bellhop returns a customer’s bags to the customer. Initial value: 0.
- **give_tip[]**: an array of semaphores where each semaphore represents the event that the customer has given a tip to the bellhop. Initial value: 0.
- **get_bellhop**: a semaphore that signals that a customer needs the assistance of a bellhop. Initial value: 0.

PSEUDOCODE

```
Void Customer(){
        Customer.bags = generateBags();
        CheckIn();
        Wait(frontdesk_ready);
        Wait(customer_exchanged);
        Signal(checked_in);
        GetRoom();
        if bags > 2 {
                GetBellhop();
                Wait(bellhop_ready);
                Wait(customer_exchanged);
                Signal(get_bellhop);
                Wait(get_bags);
        }
        EnterRoom();
        if bags > 2 {
                Signal(enters_room);
                Wait(give_bags);
                ReceiveBags();
                Post(give_tip);
        }
        Retire();
}

void Frontdesk() {
        while(true) {
                Signal(frontdesk_ready);
                Wait(checked_in);
                Signal(customer_exchanged);
                GiveRoom();
                Signal(give_room);
        }
}
```

```
void Bellhop() {
        while(true) {
                Signal(bellhop_ready);
                Wait(get_bellhop);
                Signal(customer_exchanged);
                GetBags();
                Signal(get_bags);
                Wait(enters_room);
                GiveBags();
                Signal(give_bags);
                Wait(give_tip);
        }
}
```