WAITER AGENT DESIGN DOCUMENT V2.1
* THIS IS IN ADDITION TO THE OLD DESIGN DOCUMENT

Tam Henry Le Nguyen
Tuesday Thursday
11:00-12:30
61842255   V2.1

```
Semaphore atCashier;
Semaphore receivingCheck;
Timer    relaxTimer;()
waiter State
{ Working, WantToGoOnBreak, AskedHost, CannotGoOnBreak,
  OnWayToBreak, InBreakRoom, Relaxing, WantToGoOffBreak}
boolean onBreak;
boolean requestingBreak;

Customer State
{ OutOfOrder, AskedForCheck, Getting Check,
  WaitingForCheck, CheckReceived, Given Check To
                                       Customer}
```

```
WantToGoOnBreak()
{ state = WaiterState.WantToGoOnBreak;
}
WantToGoOffBreak()
{ state = WantToGoOffBreak;
}

AllowedOnBreak(boolean answer)
{ if (answer)
     state = OnWayToBreak;
  else
     state = CannotGoOnBreak; }

CanIGetMyCheck(Customer c)
{ for (myCustomers mc : MyCustomers)
     if (mc.c == c)
        mc.state = AskedForCheck;
}

OutOfFood(int table, String food)
{ find mycustomer with table & food
     mc.state = OutOfOrder;
}
ThisIsTheCheck(Customer c, Check ch)
{ Find mycustomer that is that customer
     mc.Check = ch;
     mc.state = CheckReceived;
}
```

DATA
MESSAGES
SCHEDULER
ACTIONS

```
if (state == WantToGoOnBreak && myCustomers is empty)
   AskHost();
if (state = InBreakRoom)
   relax();
if (state = CannotGoOnBreak)
   TellHost();
if (state = WantToGoOffBreak)
   TellHost();
if (state = OnWayToBreak)
   GoOnBreak();


for (myCustomer mc : MyCustomers)
{
   if (mc.state = OutOfOrder)
      RejectCustomerOrder(mc);
   if (mc.state = AskedForCheck)
      RetrieveCheck(mc);
   if (mc.state = CheckReceived)
      DeliverCheck();
}
```

```
AskHost()
{ state = AskedHost.
  host.CanIGoOnBreak(this);
}
TellHost()
{ state = Working;
  requestingBreak = false;
  onBreak = false;
  waiter.GoToFrontDesk();
  host.BackToWork();
}

RejectCustomerOrder(myCustomer mc)
{ DoGoToCook();
  DoGoToTable();
  mc.c.OutOfChoice(mc.choice);
  mc.state = Seated;
}

RetrieveCheck(myCustomer mc)
{ DoGoTo Cashier();
  Cashier.GiveMeCheck(mc.choice, mc.c, this);
  mc.state = GettingCheck;
}
DeliverCheck(myCustomer mc)
{ DoGoToTable(mc.c);
  mc.c. HereIsYourCheck(mc.check);
```

```
relax()
{
  state=Relaxing;
  relaxTimer();
}
```