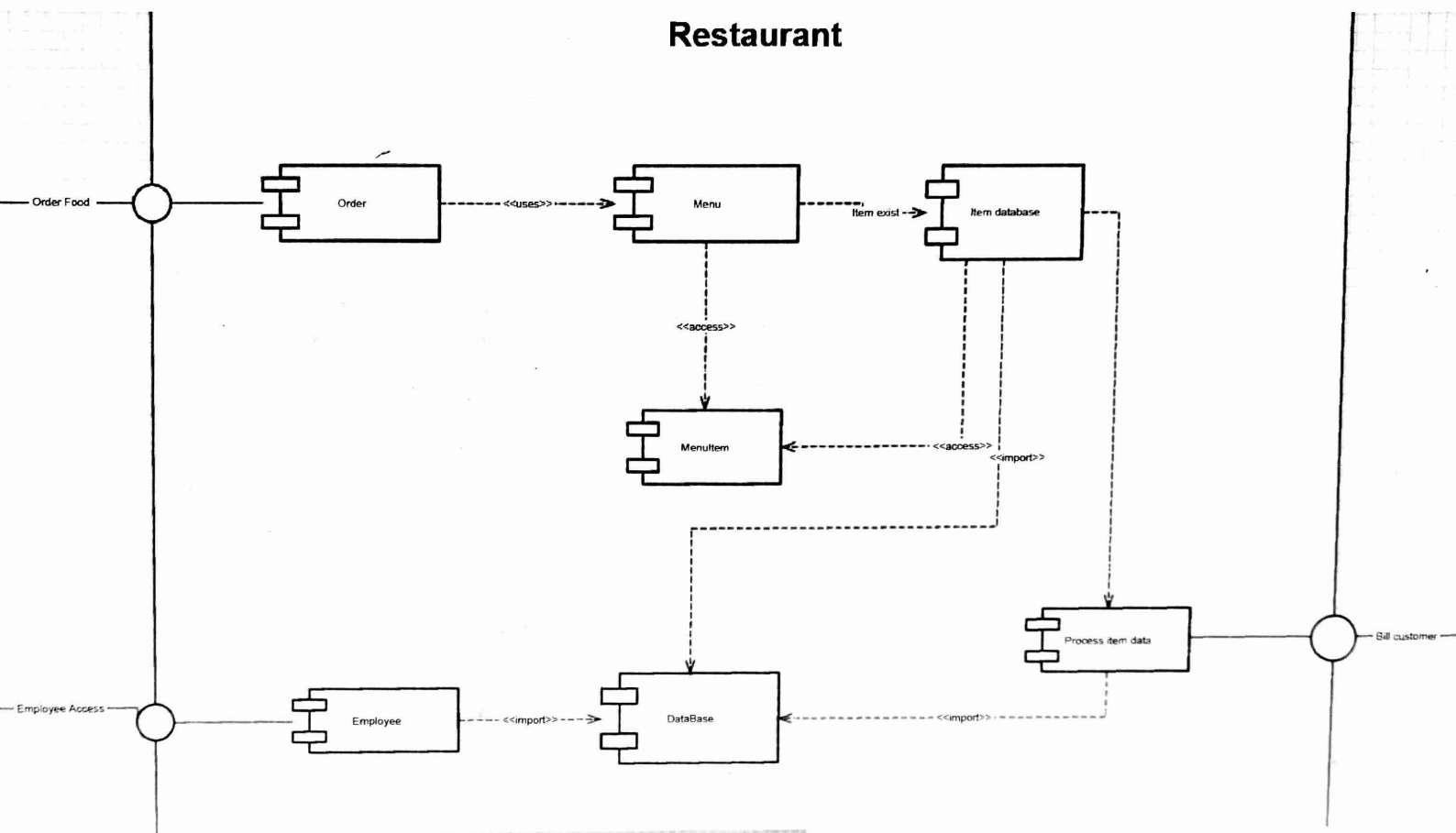


## Restaurant



## Employee Component

### Employee

- name: String, ""  
- dailyHours: decimal, 0.0  
- shiftStart: Time, 0.0  
- shiftEnd: Time 0.0

+ string getName();  
+ void clockIn();  
+ void clockOut();

clockIn():

shiftStart = clock() ;

clockOut():

shiftEnd = clock();  
dailyHours = shiftEnd - shiftStart;

# MENU COMPONENT

## Menu

- FullMenu: MenuItem

+ void addToMenu (MenuItem & item)

+ MenuItem lookup(string name)

## MenuItem

- Name : string, ""

- Price : decimal, 0.0

- Description : string, ""

- Comment: string, ""

- Next: \*MenuItem, null

+ string getName();

+ decimal getPrice();

+ string getDescription();

+ void addComment(string s);

+ MenuItem (string nme, decimal price, string descr, string comment);

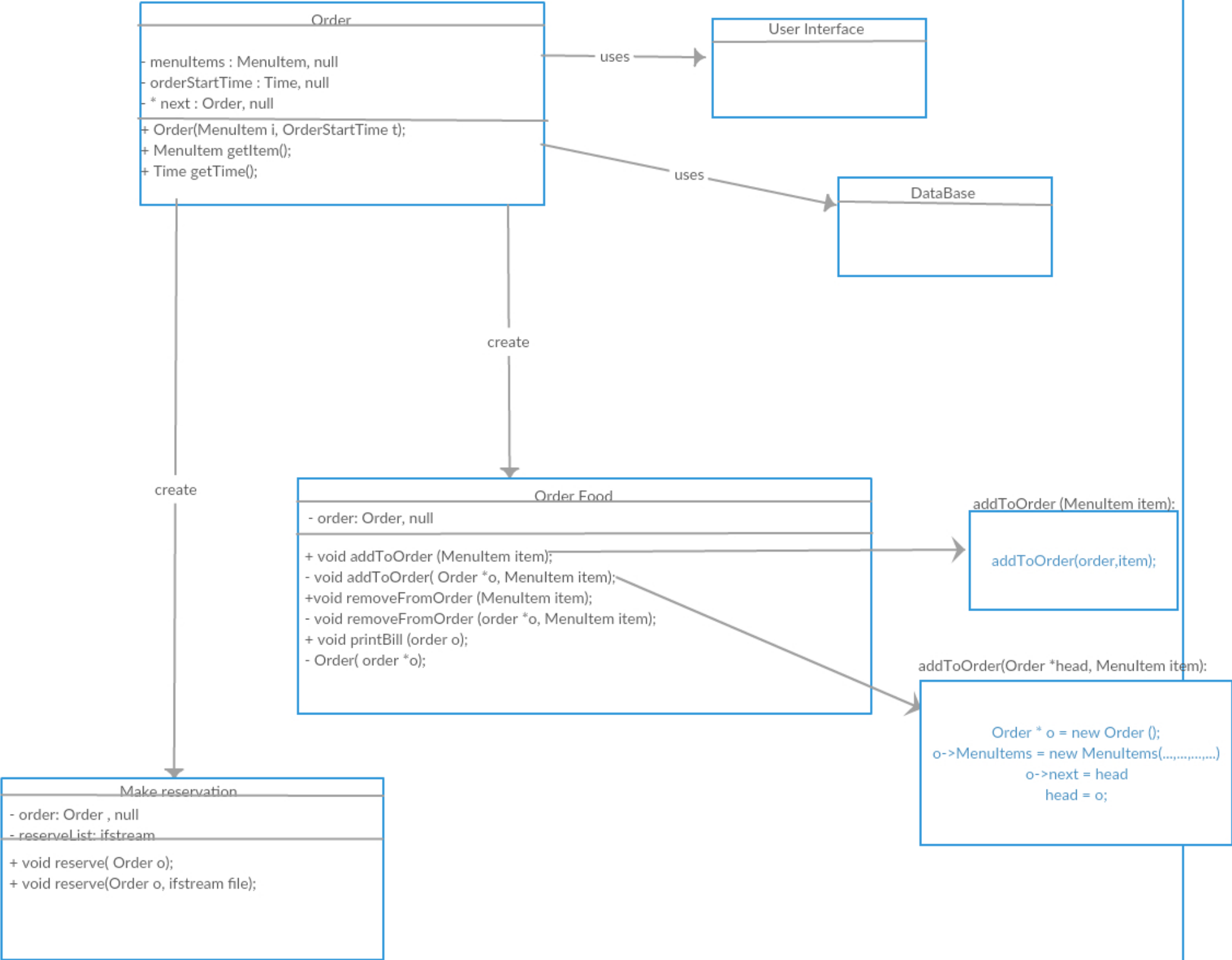
addMenu(MenuItem & item):

```
int key = item.key(item.getName(), tablesize)
list<MenuItem> & temp = table[key];
temp.push_back(item);
```

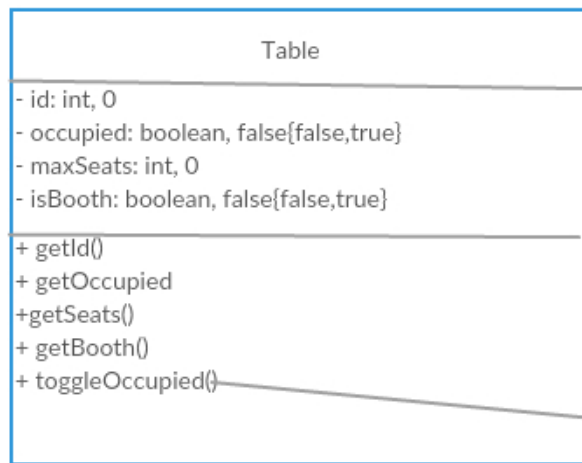
lookup(string name):

```
MenuItem i;
index = i.makeKey(name, tablesize)
if(tablesize[index].size == 0)
    return
else
    list<MenuItem>::iterator itr = table[index].begin
    while(itr != table[index].end)
        if ( *itr.getword == name)
            return *itr
```

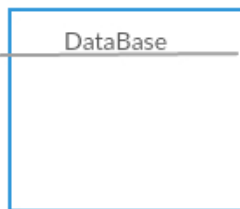
# Order Component



## Table Component



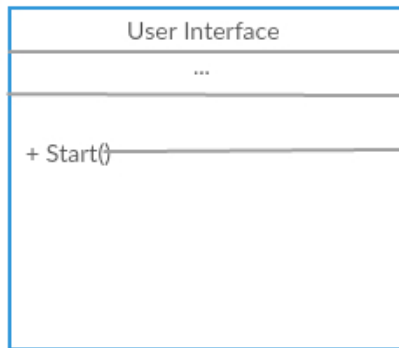
uses



toggleOccupied():

```
if occupied false
    occupied = true
else
    occupied = false
```

# User Interface



```
do{
    int option = console.readline()
    switch(option)
        case 0
            orderfood o
            o.addtoorder()
        case 1
            makesreservation r
            r.reserve
        case 2
            Employee e = new Employee(...)
            e.clockin()
            console.write("do you wish to clock out_
            int option2 = readline()
            if(option2 = 1 & isclockedin )
                e.clockout()
        case 3
            exit
    {
        while(option !=3)
```