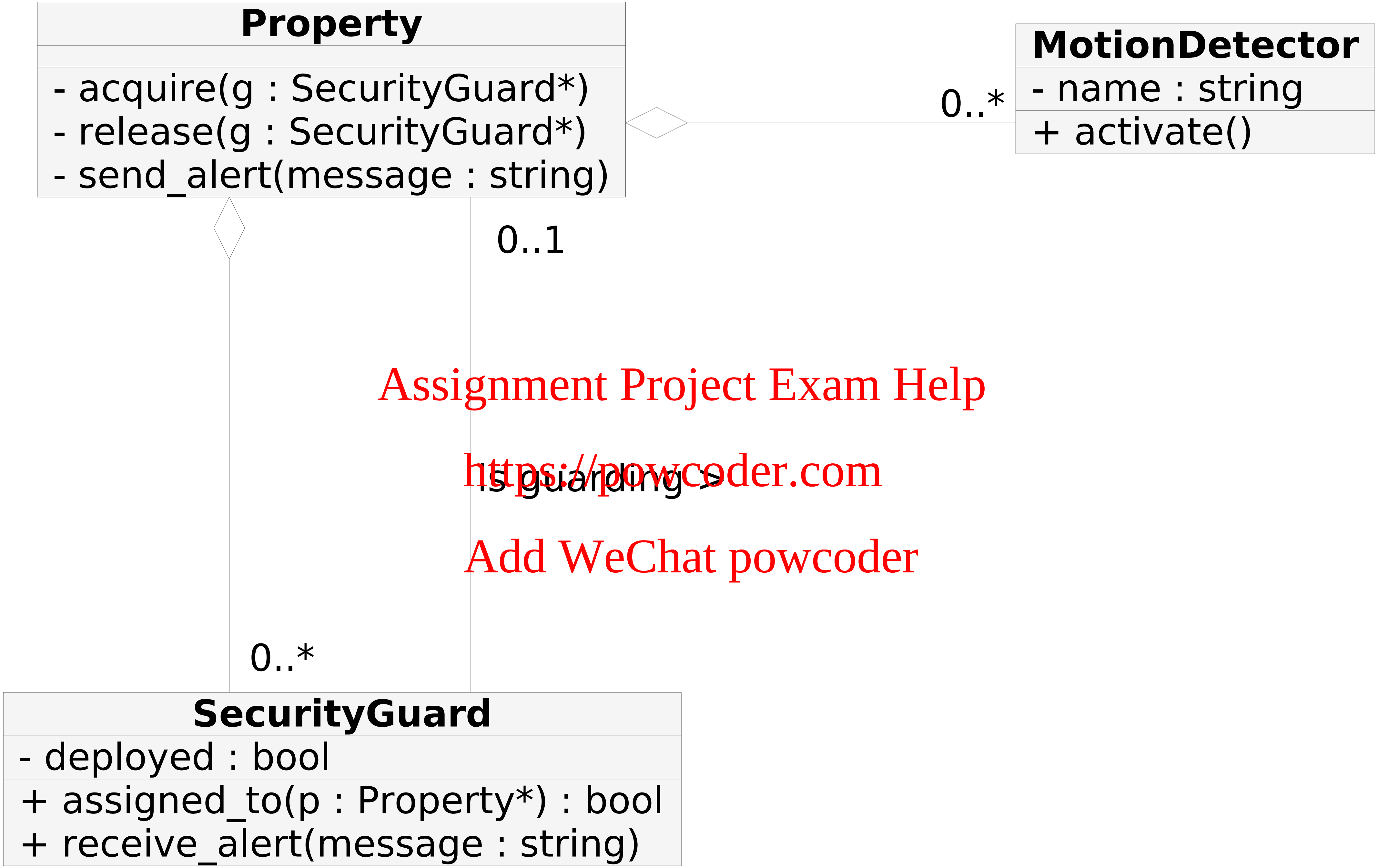
Property
- acquire(g : SecurityGuard*)
- release(g : SecurityGuard*)
- send_alert(message : string)
MotionDetector
- name : string
+ activate()
0..*
0..1
is guarding >
0..*
SecurityGuard
- deployed : bool
+ assigned_to(p : Property*) : bool
+ receive_alert(message : string)

```
#include <iostream>
#include <cassert>
#include <list>

using namespace std;

class Property;

class SecurityGuard {
private:
  bool deployed;
  Property *guarding;
public:
  SecurityGuard() {
    deployed = false;
    guarding = NULL;
  }
  bool assigned_to(Property *p);
  void receive_alert(string message) {
    cout << this << " receiving alert: " << message << endl;
  }
};

class Property {
private:
  // using STL:
  list<SecurityGuard *> guardians;
  // using given template class:
  // List<SecurityGuard *> guardians;

public:
  void acquire(SecurityGuard *g) {
    // using STL:
    guardians.push_back(g);
    // using given template class:
    // guardians.append(g);
  }
  void release(SecurityGuard *g) {
    // both STL or given template class:
    guardians.remove(g);
  }
  void send_alert(string message) {
    // using STL:
    list<SecurityGuard *>::iterator it;
    for (it = guardians.begin(); it != guardians.end(); it++)
      (*it)->receive_alert(message);
    // using given template class:
    // for (SecurityGuard **current = guardians.front(); *current; current = gua
rdians.next())
    //    (*current)->receive_alert(message);
  }
};

bool SecurityGuard::assigned_to(Property *p) {
  if (guarding)
    guarding->release(this);
  guarding = p;
  guarding->acquire(this);
  deployed = true;
  return true;
}
```

```
class MotionDetector {
private:
  Property *prop;
  string name;
public:
  MotionDetector(const char *_name, Property *_prop) {
    prop = _prop;
    name = _name;
  }
  void activate() {
    if (prop)
      prop->send_alert("Motion detected in " + name);
  }

};

int main() {

  Property kp, ic;

  MotionDetector m1("Hallway West", &kp);
  MotionDetector m2("Hallway East", &kp);
  MotionDetector m3("Crown Jewels Display Case", &kp);
  MotionDetector m4("Rector's Office", &ic);

  SecurityGuard alice, bob;

  alice.assigned_to(&kp);
  bob.assigned_to(&kp);

  m1.activate();
  alice.assigned_to(&ic);
  m3.activate();

  return 0;
```