Thomas Bergmann, 2/3/2020, Toopairs

Method: Cyclomatic Complexity

Data:

```
   344:0 updateProperty - C
   262:0 createProperty - C
   187:0 tenantRegister - B
   220:0 pmRegister - B
   120:0 pmLogin - B
   70:0 getProperty - A
   384:0 viewProperty - A
   395:0 setUpTests - A
   434:0 tearDownTests - A
   39:0 getPropertyManager - A
   57:0 getTenant - A
   86:0 addNewProperties - A
   161:0 tenantLogin - A
   174:0 login - A
   36:0 returnError - A

15 blocks (classes, functions, methods) analyzed.
Average complexity: B (5.733333333333333)
```

Analysis:

The only two methods out of normal range are the update and create property functions, which are only more complex in order to make debugging for the front-end easier. For this reason, it is not a priority to refactor these methods, but may be helpful to refactor at the end of the capstone when we will no longer have to debug the front-end and in order to have a cleaner code base. I believe our teams' CCN values are so good because most of the complex functionality of our app is handled by the Roopairs API.