CSSE 376 Software Quality Assurance
Lab 4
Congnan Zheng CM 1464

1. *mockDatabase* dynamically fakes a database connection. Then we record state of *getRoomOccupant()* as an expectation that fetching the room occupant should happen to the simulated database. And since *getRoomOccupant()* returns a value, we just compare the returned value with the expected result.

2. We can do *LastCall.Throw(somexception)*

3. Yes, it can be replaced by *DynamicMock*. If it doesn't have a return value we need to use *verify()* and *verifyAll()* to verify the behaviors.

4. *mockDatabase* dynamically fakes a database connection. Then we record state of *AvaliableRoom* as an expectation that it should return the list of rooms from the simulated database, we just compare the returned value with the expected result.

5. Since *_instance* is a private field and *Instance* only has a getter, it is not accessible in the test class. So we can use *typeof(ServiceLocator).GetField("_instance", BindingFlags.Static | BindingFlags.NonPublic).SetValue(serviceLocator, serviceLocator);* to set the field and then test the service using the regular *Assert* to check the service indeed executes the booking activity as desired .