

Ryne Bell
CM 438
CSSE 376
3/19/2012

Lab 4 answers

1. The mockDatabase object is created using a mocks.Stub object, then what it returns given a certain call is set up in the using the (mocks.Record()){ ... } block. When the Hotel object calls methods in the mockDatabase, it is returned the values we set up for the mockDatabase (so we know if the Hotel class is working properly).
2. The LastCall.Throw(Exception exception); method allows us to use LastCall to throw an exception.
3. No, we could use a DynamicMock rather than a Stub and use verify to see that the appropriate methods were called. No, we could not use a DynamicMock rather than a Stub in this case since a return value is expected, and that is what we are checking.
4. The object Rooms in the Database is set to the Rooms we created, then when the target accesses the Rooms object in the Database, it gets the Rooms object we set up.
5. We are creating a new service locator object, then overwriting the null in ServiceLocator's _instance field to make it use the ServiceLocator we created. This causes the User to access the object we created in this test case rather than a new ServiceLocator being created (As would normally happen when one has not been already created and prepopulated by the program). The removeCar method called by the target then removes the car we added to the serviceLocator, and it now has just the one we expect.