Theodore Stamp
CSSE376
3/18/2011

1. First, we create a mock object, which captures all calls that would otherwise be sent to the database object. Then, we do a record loop in which we write the code that we expect the mock object to receive, and how we would like it to respond using lastcall. Then, we verify that the code behaved as it should
2. Use LastCall.throw(exception)
3. In that case, you do not need to use a stub. Instead, you can use a dynamic mock object and verify that the calls were made as expected.
4. We are creating a mock database with an arbitrarily defined set of rooms. We then associate the database with our test hotel object. Finally, we test to see that the test object gets the available rooms from the database, and verify the value stored at rooms.
5. In this case, we have a singleton service locator that we want to juxtapose with our own service locator so that we can test it. First, we create two cars and add them to our local service locator. Then, we use reflection to replace the global variable with our local copy. Next, we remove a car and verify that it has been removed in our local instance.