Kevin Risden CM1328 3/18/11
CSSE376

Lab 4 Questions

1. The process involves creating IDatabase mock and adding it to the mocks repository. The mock is then configured to expect certain values and return certain values depending on what is being tested. After the mock is created and configured, the getRoomOccupant method is called and the result is compared to the expected value. These values, if the method is operating correctly, are returned to the method by the mock interface to the database.
2. You can tell the LastCall class to throw instead of return and then specify the exception that should be called.
3. If the mocked object did not return a value then you should use a DynamicMock. Since there is a value returned you cannot replace the stub with a DynamicMock.
4. As with the previous test a mockDatabase is of type IDatabase is added to the mocks repository created earlier. Then a List of rooms is generated and then added to the mockDatabase so that when mockDatabase.Rooms is called the list of Rooms generated is returned. Then a new Hotel object is created and the AvailableRooms method is called. The returned result is then checked against the roomCount that should exist.
5. A new service locator is created in addition to new cars. These cars are then added to the serviceLocator. Then the serviceLocator is modified so that each instance of serviceLocator is static and nonpublic so it can't be modified. Then a new User is created and books a car. Then the assertion checks that there is only 1 car left and that the car left is the one that should be the remaining car and not some random car.