Alex Jacoby

1. To mock the database, we used the white rhino framework, and since the function was returning a value, a stub was set up. Then, with the mocking stub set up, normal unit tests were run to verify that the getRoomOccupant function is working correctly.
2. Using LastCall.throw it is possible to throw an exception when a method is called.
3. If the object does not use a value you can replace it with a mock, either dynamic or nondynamic.
4. To mock the database a stub was used once again. In addition, we set the value of the mocked databases Rooms property to the local Rooms variable. Then, we checked that a hotel created had the correct number of rooms using the mocked database.
5. First a service locator and two cars are created, one car that the user will book and one that will be tested as the remaining car. These cars are added to the service locator. Then, a user is created and books the car intended for booking. Then the assertion that there is 1 remaining car is tested, and that the remaining car is the exact one that was created to be the remaining car. Both test cases pass, so we can ascertain that the service locator is working as expected.