QA IMS PROJECT
Isabella O'Hara
Testing review

Domain classes

- Here I was using JUnit to run unit tests on the domain classes (Items, Orders, OrdersItems).
- Overall I found these tests to be the easiest mostly because I was only really having to test the constructors and getters and setters which was straight forward.
- I wasn't sure how to test the 'hashcode and equals' as we've not really covered what these actually are and I tried to replicate the test in the customer domain folder for this but it would always fail – probably because my autogenerated 'hashcode and equals's' didn't look the same as the one in the Customer class.
- Learning about 'hashcode and equals' would let me improve on this.

DAO classes

- I struggled more with this mostly because my SQL syntax wasn't quite right (I had missed one comma) meaning none of the tests would run.
- Actually writing the tests I found alright partly because I could see the template for how they should look from the CustomerDAO class.
- I had lots of issues with my OrdersItemsDAO class as I kept getting the error that it couldn't find the orders_items table but I checked my SQL syntax and it was correct so I am yet (as of 5:38pm 25/08) to resolve this.

Controller classes

- Here I used Mockito unlike the other 2 where I was only using JUnit.
- I found this harder to begin with as we'd only discussed on Tuesday for the first time. But I ended up finding this easier to understand what it was doing by the end.
- I did end up having an (unresolved) issue –on my OrdersController class.
    - It was with the Update functionality => this functionality doesn't actually work which was (clearly) why this test was failing. However, this functionality wasn't included as being required for the MVP so I commented that, so anyone looking at my code could see this.
        - RE: this – update function wouldn't work here because the customer Id value that this would try to update was a foreign key to the customers table and thus couldn't be overwritten – I tried to google how to get around this but didn't have much success on finding a solution – was finding things on ENABLE UPDATE CASCADE for SQL but couldn't get that to work.

Overall thoughts

- My overall coverage come to 61.1% so beneath the industry of 80%. However this was largely in part to not having the knowledge on how to tests lots of the other classes like the IMS class, Domain class etc. meaning there were big gaps here, lowering the overall coverage.