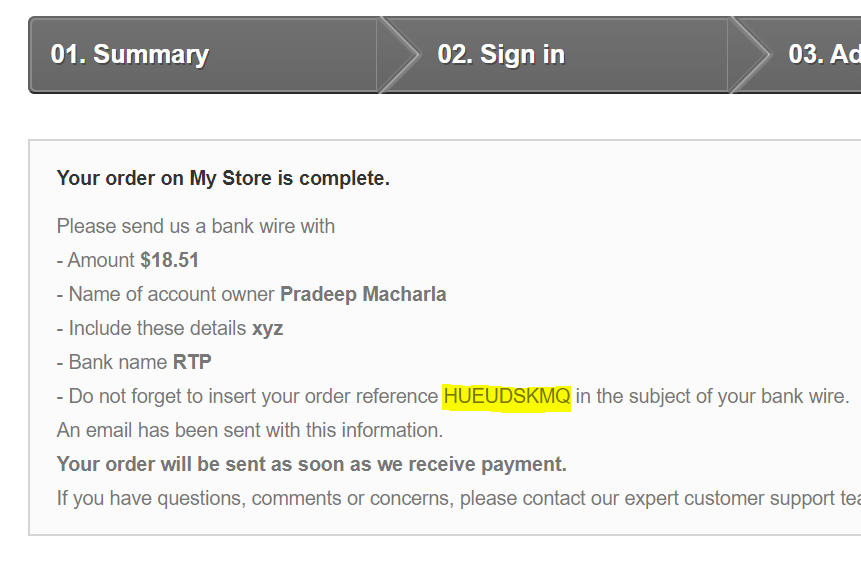
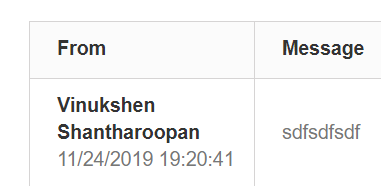
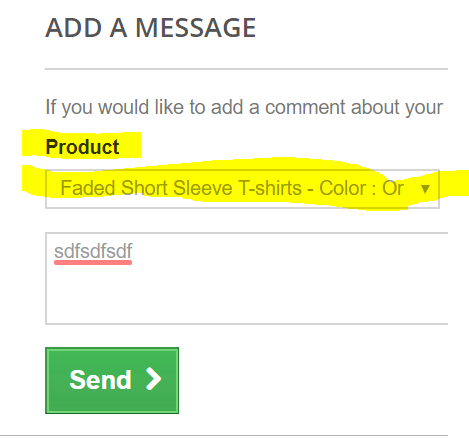
## What are the different types of testing you can perform on the web application and provide an example for each?

* Functionality Testing – testing the navigation bar works by clicking on each option in the bar and checking if they take me to the correct pages.
* Usability testing – ensuring the web application is easily visible with simple colours and the content should have no spelling or grammatical mistakes.
* Compatibility testing – ensuring the web application works on the all the available devices, i.e. Samsung S8, MacBook Pro, iPad Pro
* Performance testing – testing to see how the web application can perform in different situations. E.g. Spike, Stress and Load testing that can be tested using JMeter
* Security testing – to think like a hacker and try break into the system, i.e. view financial records.

## Did you find any defects/bugs during the testing, if so provide details and explain why you identify them as defects/bugs?

I wasn’t able to execute the tests due to a system failure, however I was able to find some bugs while surfing the website.

1. It would have been better to use ID’s instead of xpaths. Xpaths are dependent on the positioning of the elements on the webpage. If the positioning of the elements change, the xpath will also change and the automation tests will no longer be valid. However, this isn’t the case for ID’s. During my time at BT, I held a meeting with my team to ensure that every element should be given an ID. The UX/UI designers will name the ID’s while designing the elements, and these will be then passed to the developers and the testers.
2. The order reference isn’t shown clearly once the order is placed. When a customer places their order, the order reference number should appear first and in bold, however currently, it could be missed by the customer and they could lose track of their order.
3. The ‘product’ option in the message area is pointless – it doesn’t log the product anywhere as can be seen below:
4. The availability of the product is not shown at the point where the customer choses the product, and is only shown at checkout stage. This could be seen as a potential bug as there’s no warning to show that the stock is low on that specific product.



## What information you would include in a standard defect report at minimum?

Defect ID, Project name, Release version, Affected component, Build version of the product where the defect was found, Fixed build version, Summary/Description of the defect, Steps to reproduce, Actual results, Expected results, Severity, Priority, Reported by, Assigned to, Status of the Defect

## What are the key infrastructure components you would need to host the above solution in cloud (consider any cloud solution, i.e. AWS, Google, etc)?

Cloud clients – e.g. browsers, devices

Applications and Services – which supports the delivery of applications across businesses, Office365

Storage Infrastructure - cloud duplicates copies of the storage, which will benefit in the case of a disaster

Computing Infrastructure – physical and virtual resources, e.g. routers and VMware

Network - allows to connect the services over the internet

## What type of infrastructure testing you could perform to test the cloud hosted solution?

This is not just testing of the software, but also the hardware, etc. Here are a few benefits of cloud testing:

* It is very rare to replicate the environment the testers have to a customer. This may involve the lack of hardware/software infrastructure that are readily available to testers. However, this isn’t the case with Cloud testing, as both users can easily have the same environment.
* Cost effective – companies easily gather up unused hardware/software which goes to waste. For cloud users, they can only purchase parts that would be beneficial for their needs, resulting in them saving money.

Cloud testing can be categorised into four parts: testing the entire cloud, testing within the cloud, SaaS testing in the cloud and testing across different types of clouds: public, private and hybrid.

It is essential that functional and non-functional testing are covered. Functional testing must be performed to ensure the service meets the business requirements. Here are some functional tests:

* Acceptance Testing – where users are given the privileges to test cloud hosted solutions.
* Interoperability Testing – ensuring applications are working when migrating from one cloud infrastructure to another.
* System Verification Testing – to make sure components work well together within the infrastructure.

Non-functional testing is mainly performed to ensure the service meets the desired requirements. Here are some non-functional tests:

* Availability Testing – to make sure the cloud is available at all times. Intense maintenance regularly takes place, and it’s important that the administrator finds alternatives or ensure essential parts of the cloud are available.
* Performance Testing – network latency is a critical factor amongst the cloud users, thus performance testing is important.
* Security Testing – it is essential to protect sensitive information and user data integrity must always be verified.
* **Scalability Testing – measuring how application scales with increasing workload.**
* **Multi-Tenancy Testing – it is important to protect data when multiple users are interacting with a single instance.**
* Disaster recovery Testing – ensuring back-ups are easily available.

## How would you perform the different types of testing you described in question 2 above?

**I am assuming this question refers to question 5 as oppose to question 2.**

* Acceptance Testing – I would invite a mixed audience, ranging from non-technical to technical users and also business users. This will generate different and creative opinions that weren’t previously discovered by the employees, thus improving the overall quality of the infrastructure.
* System Verification Testing – firstly ensure the component-level testing has taken place and follow by incorporating retesting and regression testing in system testing.
* Performance Testing – I have the basic experience to perform stress, spike and load tests in JMeter. Ideally, I would like to further explore JMeter prior to testing the cloud hosted solution. Here are some tools that I would also look into: SOASTA CloudTest, LoadStorm, CloudTestGo, AppPerfect.

## Can you explain how you would prioritise your tests and explain if you would use any techniques?

* The test cases that are closely related to the business requirements and acceptance criteria would have the highest priority.
* The test cases that aren’t requirements but may potentially raise an issue in the future would be a medium priority.
* The test cases that would cause a small impact and aren’t important would be considered a low priority.