Black Box Testing – Search Module

1. What program unit was tested
   1. We tested the Search unit of our program
2. How we arrived at our test set data
   1. We looked at all possible SKU combinations and ran searches that would encompass boundary conditions for these possibilities
   2. This involved running tests on search possibilities and on search enter conditions
3. The actual inputs used in testing was
   1. SWEATER
   2. Sweater
   3. sweater
   4. sWeater
   5. sweaTer
   6. Black
   7. Blue
   8. Green
   9. Yellow
   10. Shirt
   11. Pants
   12. F622001
   13. S26400524564701
   14. Color
   15. “ “
   16. \\
   17. 124365478890
   18. -={}:”><
4. The expect outputs for these searches – Run 3 times each
   1. Two Items
   2. Two Items
   3. Two Items
   4. Two Items
   5. Two Items
   6. 11 Items
   7. 3 Items
   8. No Items
   9. No Items
   10. 6 Items
   11. 4 Items
   12. 1 Item
   13. 1 Item
   14. No Items/No Crash
   15. No Items/No Crash
   16. No Items/No Crash
   17. No Items/No Crash
   18. No Items/No Crash
5. Variables, Files, States
   1. String 🡪 JavaReader.txt
   2. JavaReader.txt 🡪 Java
   3. Java 🡪 Database
   4. Database 🡪 Java
   5. Java 🡪 HTMLReader.txt
   6. HTMLReader.txt 🡪 String
   7. Displayed
6. Testing performed by KiviuqSoftware™ a full stack development group
7. No discrepancies between expected outputs and actual outputs in test runs, troubleshooting done primarily in IDE so by testing the software was building and running error free in a contained local environment

Glass Box Testing – Search Module

1. What program unit was tested
   1. We tested the Search unit of our program
2. How we arrived at our test set data
   1. We looked at all the code and investigated to see what potential validation errors could be thrown and went in with an intention to break the logic in our system
   2. This involved running tests on search possibilities and on search enter conditions
3. The actual inputs used in testing was
   1. Sweater Jacket
   2. Black Green Purple
   3. SS245454654
   4. Break this
   5. Filter me I dare you
   6. This is how we do
   7. hhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh
   8. when in doubt
   9. New Yourk CIty
   10. Madon
   11. Awake
   12. S524564701
   13. Color Type Stlye
   14. Dresses
   15. -Black--
   16. Partial amtch
   17. Bla gr ye ye ye
4. The expect outputs for these searches – Run 3 times each
   1. No Items/No Crash
   2. No Items/No Crash
   3. No Items/No Crash
   4. No Items/No Crash
   5. No Items/No Crash
   6. No Items/No Crash
   7. No Items/No Crash
   8. No Items/No Crash
   9. No Items/No Crash
   10. No Items/No Crash
   11. All Items
   12. No Items/No Crash
   13. No Items/No Crash
   14. No Items/No Crash
   15. 6 items
   16. No Items/No Crash
   17. No Items/No Crash
5. Variables, Files, States
   1. String 🡪 JavaReader.txt
   2. JavaReader.txt 🡪 Java
   3. Java 🡪 Database
   4. Database 🡪 Java
   5. Java 🡪 HTMLReader.txt
   6. HTMLReader.txt 🡪 String
   7. Displayed
6. Testing performed by KiviuqSoftware™ a full stack development group
7. One discrepancy found between expected outputs and actual outputs in test runs, troubleshooting done primarily in IDE so by testing the software was building and running error free in a contained local environment. Validity checking was added to fix an error where symbols added with a valid search string returned no objects rather than objects matched with valid string.