**Prioritizing the tests to avoid dependency issues:**

If you observe the report, some of the test cases failed because they have dependency on each other.

For example, we adding 1 vehicle in one test case and checking that the no. of vehicles on dashboard are increased by one and in another test cases we are adding 2 vehicles and checking that the no. of vehicles on dashboard are increased by two. If these two test cases are executed parallelly both test cases fail because they add by the time the first test case completes, the second test case adds a vehicle or two so the difference would be more than 1. Similarly for the second test case since the first test case already adds a vehicle, the difference would be more than two. So both the test cases fail.

If we have the tests as separate methods, then we can use dependency attributes to avoid this. For example, we can say that the second test case is dependent on first test so it won’t be executed until the second one is completed. But in keyword driven framework, we can’t do this as there is only one method. So how to do this?

We can have priority for the test cases. Add a column main test cases sheet with column name priority and give priority to all test cases. Make sure that you are giving different priority based on the dependency I have explained earlier.

Please check the main test cases excel sheet for the priority I have given.

Now when we read these test cases, we need to consider this priority and execute the test cases as per the priority. So once we read all the test cases (test case, workbook and sheet name) we read the priority also and sort the lists based on the priority.

So in the ExcelUtilities class we modify the method getTestCases() a little.

//This function will return all the main test cases like TC01, TC02 etc where run = "yes"

**public** List<List<String>> getTestCases(String wbpath, String sheetname) **throws** IOException{

//This list holds all the test cases

List<List<String>> testcases = **new** ArrayList<List<String>>();

FileInputStream fis = **new** FileInputStream(wbpath);

XSSFWorkbook workbook = **new** XSSFWorkbook(fis);

XSSFSheet sheet = workbook.getSheet(sheetname);

//Get all the rows

Iterator<Row> rows = sheet.iterator();

**while**(rows.hasNext()) {

//This list holds each test case info test case no, workbook, sheet

List<String> testcaseinfo = **new** ArrayList<String>();

Row rowno = rows.next();

Cell tcvalue = rowno.getCell(0);

Cell wbvalue = rowno.getCell(1);

Cell shvalue = rowno.getCell(2);

Cell rvalue = rowno.getCell(4);

**Cell prvalue = rowno.getCell(5);**

String testcasevalue;

String workbookvalue;

String sheetvalue;

String runvalue;

**String priorityvalue;**

DataFormatter df = **new** DataFormatter();

runvalue = df.formatCellValue(rvalue);

**if**(runvalue.equalsIgnoreCase("yes")) {

testcasevalue = df.formatCellValue(tcvalue);

testcaseinfo.add(testcasevalue);

workbookvalue = df.formatCellValue(wbvalue);

testcaseinfo.add(workbookvalue);

sheetvalue = df.formatCellValue(shvalue);

**priorityvalue = df.formatCellValue(prvalue);**

testcaseinfo.add(sheetvalue);

**testcaseinfo.add(priorityvalue);**

testcases.add(testcaseinfo);

}

}

workbook.close();

**testcases.sort((l1, l2) -> l1.get(3).compareTo(l2.get(3)));**

**return** testcases;

}

First I am reading each of the priority values and adding it to each list. Then I am adding that list to the main list testcases. Now I am sorting the main list testcases based on the priority value in each list.

**testcases.sort((l1, l2) -> l1.get(3).compareTo(l2.get(3)));**

Now the test cases will be sorted based on the priority value and if you run the test cases now all of the test cases will pass.