# Report: Project 1: Specification-Based Testing (Part 2 -Design of Experiments)

Nigar Sultana

ID: 1221371489

### Explanation of the test cases

In this project, we are asked to utilize the Design of Experiments (DOE) methodology for generating pairwise combination tests as per the provided testing specifications. In order to achieve this, I analyzed various DOE tools to identify the most appropriate ones. Subsequently, I chose to add the outcomes using two distinct tools within the test case screenshot section. Although Both testing tools covered all possible test cases, but the first one is optimal as it has 25 test cases, which is less compared to the 35 test cases but covers all the required test cases. Here, I have explained 25 test cases that I got from the first testing tool.

All five types of phones (iPhone X, iPhone 8, Samsung S9, Huawei Mate, and Google Pixel 3) are paired with both parallel tasks running (Yes, No).

All five types of phones (iPhone X, iPhone 8, Samsung S9, Huawei Mate, and Google Pixel 3) are paired with all connectivity (Wireless, 3G, 4G, Edge).

All five types of phones (iPhone X, iPhone 8, Samsung S9, Huawei Mate, and Google Pixel 3) are paired with all memories (1GB, 2GB, 4GB,6GB).

All five types of phones (iPhone X, iPhone 8, Samsung S9, Huawei Mate, and Google Pixel 3) are paired with all battery levels (<20 %, 20-39 %, 40-59 %, 60-79 %, 80-100 %).

Both parallel tasks running (Yes, No) are paired with all connectivity (Wireless, 3G, 4G, Edge).

Both parallel tasks running (Yes, No) are paired with all memories (1GB, 2GB, 4GB, 6GB).

Both parallel tasks running (Yes, No) are paired with all battery levels (<20 %, 20-39 %, 40-59 %, 60-79 %, 80-100 %).

All connectivity (Wireless, 3G, 4G, Edge) are paired with all memories (1GB, 2GB, 4GB, 6GB).

All connectivity (Wireless, 3G, 4G, Edge) are paired with all battery levels (<20 %, 20-39 %, 40-59 %,60-79 %, 80-100 %).

All memories (1GB, 2GB, 4GB, 6GB) are paired with all battery levels (<20 %, 20-39 %, 40-59 %,60-79 %, 80-100 %).

**Screenshot of the report:**

| | Type of Phone | Parallel Tasks Running | Connectivity | Memory | Battery Level |
|---|---|---|---|---|---|
| 1 | iPhone X | Yes | Wireless | 1 GB | < 20 % |
| 2 | iPhone X | No | 3G | 2 GB | 20 - 39% |
| 3 | iPhone X | Yes | 4G | 4 GB | 40 - 59% |
| 4 | iPhone X | No | Edge | 6 GB | 60 - 79% |
| 5 | iPhone X | Yes | Wireless | 1 GB | 80 - 100% |
| 6 | iPhone 8 | No | 4G | 6 GB | 80 - 100% |
| 7 | iPhone 8 | Yes | Edge | 1 GB | < 20 % |
| 8 | iPhone 8 | No | Wireless | 1 GB | 20 - 39% |
| 9 | iPhone 8 | Yes | Wireless | 2 GB | 40 - 59% |
| 10 | iPhone 8 | Yes | 3G | 4 GB | 60 - 79% |
| 11 | Samsung S9 | Yes | Wireless | 2 GB | 60 - 79% |
| 12 | Samsung S9 | No | Wireless | 4 GB | 80 - 100% |
| 13 | Samsung S9 | Yes | 3G | 6 GB | < 20 % |
| 14 | Samsung S9 | Yes | 4G | 1 GB | 20 - 39% |
| 15 | Samsung S9 | No | Edge | 1 GB | 40 - 59% |
| 16 | Huawei Mate | No | 3G | 1 GB | 40 - 59% |
| 17 | Huawei Mate | Yes | 4G | 1 GB | 60 - 79% |
| 18 | Huawei Mate | Yes | Edge | 2 GB | 80 - 100% |
| 19 | Huawei Mate | No | Wireless | 4 GB | < 20 % |
| 20 | Huawei Mate | Yes | Wireless | 6 GB | 20 - 39% |
| 21 | Google Pixel 3 | Yes | Edge | 4 GB | 20 - 39% |
| 22 | Google Pixel 3 | Yes | Wireless | 6 GB | 40 - 59% |
| 23 | Google Pixel 3 | No | Wireless | 1 GB | 60 - 79% |
| 24 | Google Pixel 3 | Yes | 3G | 1 GB | 80 - 100% |
| 25 | Google Pixel 3 | No | 4G | 2 GB | < 20 % |

Figure: 1.1 Screenshot of 25 test cases created by the first tool.

| | Type of Phone | Parallel Tasks Running | Connectivity | Memory | Battery Level |
|---|---|---|---|---|---|
| 1 | iPhone 8 | Yes | Wireless | 2 GB | 80-100% |
| 2 | Huawei Mate | Yes | Wireless | 4 GB | 20-39% |
| 3 | Huawei Mate | No | 3G | 1 GB | 40-59% |
| 4 | Google Pixel 3 | No | Edge | 4 GB | 60-79% |
| 5 | iPhone X | Yes | Edge | 1 GB | 80-100% |
| 6 | Google Pixel 3 | Yes | Wireless | 2 GB | 60-79% |
| 7 | iPhone 8 | Yes | 4G | 4 GB | <20 % |
| 8 | iPhone 8 | Yes | Edge | 6 GB | 60-79% |
| 9 | Samsung S9 | No | Wireless | 1 GB | <20 % |
| 10 | Samsung S9 | No | Edge | 2 GB | <20 % |
| 11 | iPhone X | No | 4G | 1 GB | 40-59% |
| 12 | iPhone X | Yes | Wireless | 4 GB | 80-100% |
| 13 | Samsung S9 | Yes | 3G | 6 GB | 80-100% |
| 14 | iPhone 8 | No | 4G | 1 GB | 20-39% |
| 15 | Huawei Mate | Yes | 4G | 4 GB | 40-59% |
| 16 | Samsung S9 | No | 4G | 2 GB | 40-59% |
| 17 | Huawei Mate | No | Edge | 2 GB | 60-79% |
| 18 | iPhone X | No | 4G | 2 GB | 80-100% |
| 19 | Google Pixel 3 | No | Edge | 1 GB | 20-39% |
| 20 | iPhone 8 | No | 3G | 6 GB | 40-59% |
| 21 | iPhone X | Yes | 3G | 2 GB | 60-79% |
| 22 | Huawei Mate | No | 3G | 4 GB | <20 % |
| 23 | Samsung S9 | No | Wireless | 6 GB | 20-39% |
| 24 | Samsung S9 | No | 3G | 4 GB | 20-39% |
| 25 | Huawei Mate | No | 4G | 6 GB | 60-79% |
| 26 | Google Pixel 3 | Yes | 3G | 1 GB | 60-79% |
| 27 | Google Pixel 3 | No | Edge | 6 GB | 40-59% |
| 28 | Google Pixel 3 | No | 4G | 6 GB | <20 % |
| 29 | iPhone X | No | 3G | 6 GB | 20-39% |
| 30 | Huawei Mate | Yes | 4G | 2 GB | 20-39% |
| 31 | Samsung S9 | Yes | Wireless | 2 GB | 60-79% |
| 32 | Huawei Mate | No | 3G | 6 GB | 80-100% |
| 33 | iPhone X | Yes | Wireless | 4 GB | <20 % |
| 34 | Huawei Mate | Yes | Wireless | 1 GB | 40-59% |
| 35 | Google Pixel 3 | Yes | Wireless | 6 GB | 80-100% |

Figure: 2.1 Screenshot of 35 test cases created by the second tool.

**An assessment of the test cases**

The tool-generated test case effectively covers a wide range of parameter combinations. Each test case represents a unique combination of values for “Type of Phone”, “Parallel Tasks Running “, “Connectivity”, “Memory”, and “Battery Level”. This comprehensive coverage ensures that interactions among different parameters are tested thoroughly.

The test case follows the principles of DOE (Design of Experiments) by generating test cases that optimize coverage while minimizing the number of test cases. Instead of testing every possible combination, the tool intelligently selects combinations that prove the most coverage of

interactions. For instance, we have five parameters in this project each of which contains distinct values. The parameter “Type of Phone” contains five values: iPhone X, iPhone 8, Samsung S9, Huawei Mate, and Google Pixel 3. Likewise, the parameter “Parallel Tasks Running” offers two options: Yes and No. Similarly, the parameter “Connectivity” presents four choices: Wireless, 3G, 4G, and Edge. Moreover, the “Memory” parameter introduces values like 1 GB, 2 GB, # GB, 4GB, and 6GB, while the “Battery Level” parameter comprises five ranges: <20%, 20-39%, 40-59%, 60-79%, and 80-100%.

To thoroughly assess all possible combinations of input values without using this tool, a total of 5 * 2 * 4 * 4 * 5 = 800 test cases would be needed. However, applying a pairwise combination testing approach can bring this number down to only 25 while evaluating every possible pair of input values. This methodology helps to reduce the time and resources required for the testing process.

The test cases reflect real-world usage scenarios, as they involve combinations of attributes that users might encounter while using the mobile application.

In a nutshell, the test cases generated by the tool align with the principles of the Design of Experiments (DOE) technique.

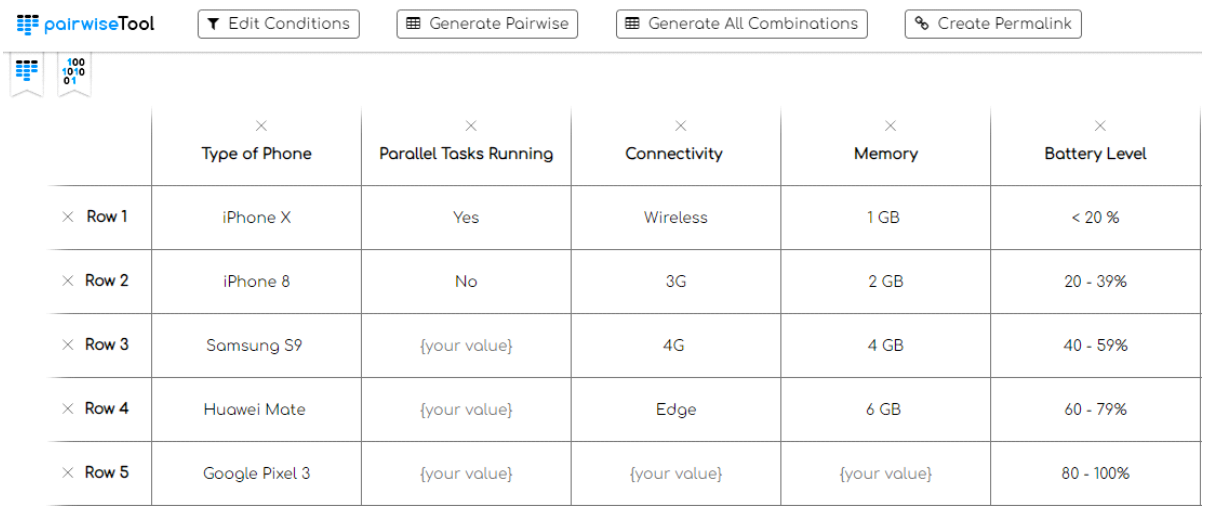
**An assessment of the tool**

First Tool:

- Features and functionalities:

Web-based pairwise tool is the first tool that offers features that allow users to define conditions, generate optimized test cases, create exhaustive combinations, and share configurations using permalinks. For example, “Edit Conditions “allows users to define and customize the conditions for generating test cases, “Generate Pairwise “initiates the process of creating pairwise combinations of parameter values for generating test cases.

To work with this tool, I added the parameters required for example 1. Type of Phone 2. Parallel Tasks Running 3. Connectivity 4. Memory, and 5. Battery Level as the columns by clicking on the “+” button. In the same way, the values for example in Row 1 iPhone X, yes, Wireless, 1 GB, <20%, and for Row 2 iPhone 8, No, 3G, 2 GB, 20-39, and so on were added for each parameter. Below Figure 1 shows the table of input parameters and their respective values that were used to create the pairwise test cases.




| | Type of Phone | Parallel Tasks Running | Connectivity | Memory | Battery Level |
|---|---|---|---|---|---|
| Row 1 | iPhone X | Yes | Wireless | 1 GB | < 20 % |
| Row 2 | iPhone 8 | No | 3G | 2 GB | 20 - 39% |
| Row 3 | Samsung S9 | {your value} | 4G | 4 GB | 40 - 59% |
| Row 4 | Huawei Mate | {your value} | Edge | 6 GB | 60 - 79% |
| Row 5 | Google Pixel 3 | {your value} | {your value} | {your value} | 80 - 100% |

Figure 1. Parameters and respective values (for the first tool).

When the “Generate Pairwise” button is pressed, the tool makes a list with 25 sets of tests. These tests have all the different pairs of parameter values. The output of this tool is shown in the above figure 1.1(Screenshot of test cases section).

- Area of usage:

  This tool is mainly used in generating efficient and optimized test cases using the pairwise combination technique. This technique is especially valuable when comprehensive testing of parameter interactions are required, but the number of test cases needs to be minimized.

- Ease of use:
  This tool’s grid format allowed me to easily input parameter values in rows and columns. In order to smoothly generate the test cases this tool also introduces a user-friendly interface.
- Ease of setup and install:

  One of the big advantages of this tool is I didn’t need to install or set up any software on my local machine. I got this tool directly by accessing this link: https://pairwise.teremokgames.com through my Chrome browser.

Second Tool:

- Features and functionalities:
  There are examples like Export to JIRA and Export to Excel where users can export test cases as JIRA problems and an Excel sheet, respectively. Using the generated test cases, we can generate test scripts that can be given to testers, who can then incorporate them into testing tools, or add them to code bases as automated tests. This tool also provides conditional test scripts, which are dependent on the value of the test cases. They claim to have the fastest pairwise algorithm on the market.

  In this second tool, I defined the parameters by selecting “Define Parameters “option and providing the respective values associated with the required parameters are shown below in Figure 2:

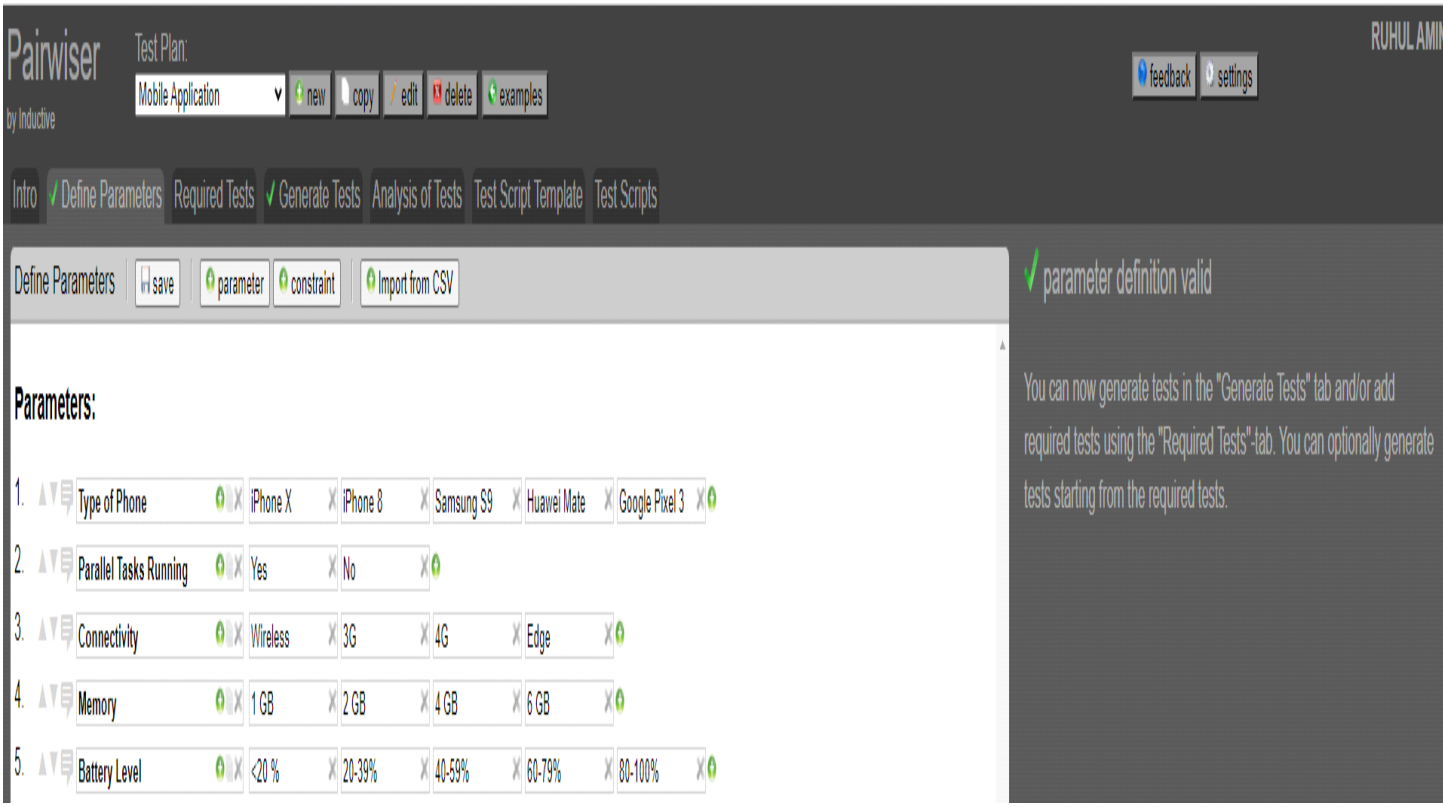



Figure:2 “Pairwiser “values associated with the respective parameters

Once the previous steps are saved, the “generate tests” button which is located under the “Generate Tests” tab is clicked to generate the test cases. This tool generated 35 test cases which covered all possible pairwise combinations. Figure 2.1 (Screenshot of test cases section) shows the output created by this tool. After that to analyze the tests, I clicked on the analyze button under the “Analysis of Tests “Tab. A screenshot of this test case analysis is shown in Figure 3 below.

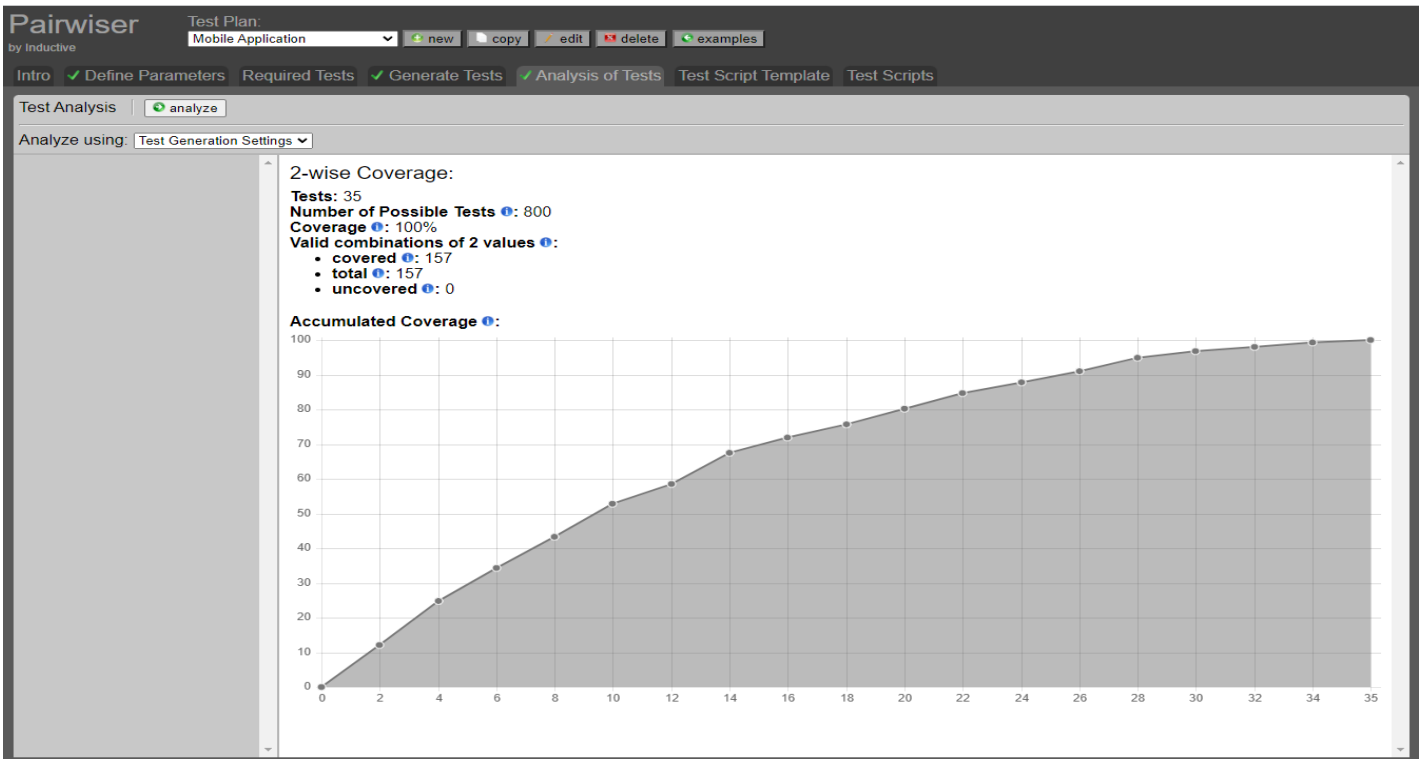



Figure: 3 Accumulated Coverage using second testing tool.

- Area of usage:

Pairwiser is a free tool which is utilized to create pairwise combinations. Besides, it can be used to test desktop applications, mobile applications, websites, a product line, or a highly configurable system.

- Ease of use:

Users can easily understand Pairwiser's user interface due to its simplicity. This straightforward tool gives the user the chance to create 1-wise, 2-wise, 3-wise, or mixed test cases by defining parameters with their values, constraints, and required tests.

- Ease of setup and install:
  The second DOE tool that I have used is "Pairwiser "which I got by creating an account on this website https://inductive.no/pairwiser-tool/. After" sign in "in this Pairwiser account I found it easily. I didn't face any installation and setup issues.