Simulation tool for computing platform

TEST RESULTS

Version <1.0>

TABLE OF CONTENTS

1 Introduction

1.1 Purpose of The Test Results Document

This Test results document shows all the results of the tests planned in the test plan document. Its intended audience is the IT support team.

2 Compatibility Testing

2.1 Test Recall

The principal compatibility problems we can encounter with this kind of software (web-application) is the web browser compatibility, since browsers are already compatible on every kind of computing environment.

2.2 Test Results

Tested page	Google Chrome (v 63)	Mozilla Firefox (54.0.1)	Safari (11.0.2)	Internet Explorer
Simulation parameters	OK	OK	OK	Not tested
Running simulation & output	OK	OK	OK	Not tested

This test is passed for most common browsers, however, errors could happen on internet explorer (the reason why I have not tested with IE is that I work with an iOS laptop and there is no support for IE on a mac).

3 Conformance/Functional Testing

3.1 Test Recall

These tests will confirm all the features announced in the requirements.

3.2 Test Results

All specifications are working. All tests have passed.

4 Performance/Load Testing

4.1 Test Recall

The idea of those tests is to compare the performance of the software under differents work loads.

4.2 Test Results

We can see a good behaviour under big loads

Output

```
-------- SIMULATION RESULTS -------

Simulation duration : 5 week(s).

Number of nodes : 1280
Number of cores : 16
Total number of cores : 20480

Number of students classes : 10
Total number of students : 200

Number of researchers groups : 5
Total number of researchers : 25


 ------- Week 1 -------

Amount of small jobs performed : 3817
Amount of medium jobs performed : 171
Amount of large jobs performed : 14
Amount of huge jobs performed : 2


 ------- Week 2 -------

Amount of small jobs performed : 1102
```

We can also see that results are proportionals to input values. For example, adding 10 times more nodes will give an amount of jobs performed by week 10 times bigger.

5 Unit Testing

Unfortunately, unit testing with Karma for Angular Components (outside HTML DOM elements tests like previous ones) requires some skills I do not have, and some formation time I do not have too. Moreover, I don't think there is some more testing tools for Angular.