**Test Summary Report**

**Purpose**

This document is to summarise the various activities undertaken when testing the Herokuapp Computer database application. This document will cover the tests executed, bugs and issues found as well as provide feedback on the system under test.

**Application Overview**

The Herokuapp Computer Database is an web based database application. Users are able to add, maintain and delete entries as required as well as filter out data specific for their needs. The database is updated in real-time with every change that the user makes to either existing or new computer entries made, or removal of existing ones.

**Testing Scope**

**In Scope:**

The testing which was performed in scope as part of this task was around the CRUD (Create, Read, Update, Delete) operations as well as the systems edge cases

**Out of Scope:**

Testing was not performed against

* Performance
* User Interface/Experience

**Metrics**

1. Number of test cases planned vs executed
2. Number of test cases passed vs failed

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Cases Planned** | **Test Cases Executed** | **Tests Passed** | **Tests Failed** |
| 47 | 47 | 43 | 4 |

1. Number of issues raised: 4
2. Number of bugs found: 5

**Types of Testing used**

Testing which was performed against the system covered the CRUD operations as well as the edge cases that surround those operations. As there was no pre-existing requirements to work with, the approaches taken were predominately exploratory and ad-hoc testing.

**Test Environment**

The system under test can be found on this URL <http://computer-database.herokuapp.com/computers>

**Key Learnings and Findings**

During my testing there were a few key learnings and findings. Firstly, understanding the scope of testing was crucial. Initially in my planning I was focusing on the entirety of the system rather than focusing on the task at hand (CRUD operations and edge cases). As I re-evaluated my approach I found I was able to methodologically flow through the system workflow focusing on each operation and proceeding through the steps of the system.

Another key learning point was that I was getting a little to caught up in the detail of some of the bugs found. I found that i had identified the problem, but I then went into the mindset of attempting to work out the root cause and possible solution to fix the problem (eg: the sorting of the table).

Some key findings were that the system behaved differently when there was either just text, numbers or special characters used in the computer name but also using them in combination also had adverse and interesting results which are documented in greater detail in the spreadsheet provided.

**Recommendations and Conclusion**

Having focused only on the CRUD operations and edge cases, the system is still at risk of introducing other bugs in areas not properly tested. Having no conditions on field length also caused the system to crash when a large amount of text was entered into the computer name and then attempting to save this to the database. Some basic validation controls around field length, allowable characters and general system behaviour would provide a more robust and economical system. My advice and conclusion is that the system needs to have further testing conducted with a refined view on the system requirements being identified to allow for a quicker testing turnaround and testing cycle times.