CHAPTER 7

CONCLUSION

From the overall testing and analysis of PReWebD and PReWebN we can make the following conclusion.

i. In our investigations it is found that the PReWebD implemented with .NET techniques is capable of handling up to 40 concurrent users smoothly. The PReWebN implemented with Java techniques is capable of handling up to 50 concurrent users without any error. As such we can conclude that the Web application implemented with Java technique is more scalable than the Web application implemented with .NET technique.

ii. If we observed the recorded data of testing on insert operation, the PReWebD implemented with .NET technique performs much faster than the PReWebN implemented with Java technique considering the same set of data. For delete operation also, we found the same results. The PReWebD returns more throughputs to the client than PReWebN. As such, we can conclude that the Web application implemented with .NET technique is more efficient than the Web application implemented with Java technique.

iii. The statistical analysis of data recorded for PReWebD and PReWebN shows that both .NET and Java techniques are stable.

iv. The statistical analysis of data recorded for PReWebD and PReWebN also shows that both .NET and Java techniques are reliable.

v. The PReWebD is implemented with proprietary software tools while the PReWebN is implemented with open source software tools. As such Java technique is more cost effective than .NET technique.

vi. For PReWebD implemented with .NET technique and PReWebN implemented with Java technique, hits/sec and throughput have individual as well as combined effect on response time.

As such from our overall investigations it can be concluded that if we consider factors, e.g. scalability, stability, reliability and cost, then the performance of Java
technique is far better than its counterpart .NET technique. But, if we consider the efficiency, then .NET technique overtake its counterpart Java technique.

The drawbacks of the present work may be overcome by including the following:

a. The PReWebD and PReWebN may be implemented with large datasets to more accurately benchmark the two techniques.

b. Recording replica of data for each test and analyzing those data may also provide better insight of the two techniques.

c. Rigorous testing and analysis of PReWebD and PReWebN are necessary to have an in-depth idea of the factors hampering the Web applications developed with both the techniques.

d. A suitable method should be implemented to remove the garbage collected heap during the testing of both the applications and analysis of those recorded data may also help for accurate benchmarking of the two techniques.