**Scalability**

**Expectation**

Firstly, the system should be able to handle all registered students, lecturers, tutors and teaching assistants of the COS department. It should be possible to add more students as required.

**Prioritization**

Critical

**Example in System**

Handling an increasing number of posts from an increasing amount of students.

**Performance Requirements**

**Expectation**

The system should not spend a lot of time on processing infrasture code, so that additions and deletions to the Buzz Space can almost happen immediately.

**Prioritization**

Nice-to-have

**Example in System**

Posts/Comments should appear/deleted almost instantaneous after the command has been sent (within a second).

**Reliability and Availability**

**Expectation**

System failure due to bugs and other architectural problems are kept to a minimum.

**Prioritization**

Critical

**Example in System**

1 in 1 million transactions fail due to architectural problems.

**Security**

**Expectation**

Thorough authentication should be done before giving users access or providing services like posting and deleting posts.

All information should be kept confidential.

**Prioritization**

Critical

**Example in System**

Logging in should be done and some sort of verification before a big action like a deleting a thread.

**Testability**

**Expectation**

Create software that is easy to test in various test contexts.

**Prioritization**

Nice-to-have

**Example in System**

The login system can be tested and faults can be found easily.

**Integrability**

**Expectation**

Integration should be seamless because sophisticated integration channels and protocols are used.

**Prioritization**

Critical

**Example in System**

Combining the authorization subsystem with the threads subsystem in order to create a part of a BuzzSpace system.