**Scalability**

**Stakeholders**

The stakeholders for this requirement are: Programmers involved in System maintenance.

**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, by the addition of system resources, but without making changes to the deployment architecture.

**Prioritization**

Critical

**Strategy**

Make reasonable predictions about projections of expected growth based on the BuzzSpace requirements and using data from previous years.

The design should also be flexible enough to handle increased loads until a system can be upgraded with additional resources.

**Example in System**

Handling an increasing number of posts from an increasing amount of students in the years to come.

**Design Patterns**

Flyweight can also be used minimizes memory use by sharing as much data as possible with other similar objects.

Loose coupling and Parallelism can also be used to improve the scalability of the system.

**Integrability**

**Stakeholders**

The stakeholders for this requirement are: Programmers involved in System creation.

**Expectation**

Integration of the different subsystems (BuzzSpace, Authorization, Services etc) should be seamless because sophisticated integration channels and protocols will be used. This will ensure that the system as a whole can will be composed effectively via the different subsystems.

**Prioritization**

Critical

**Strategy**

**Example in System**

The BuzzSpace has to be integrated into the CS department’s website so that the user’s credentials from the CS LDAP can be used for authentication.

**Testability**

**Stakeholders**

The stakeholders for this requirement are: Programmers involved in System creation.

**Expectation**

Create software that is easy to test in various test contexts. The system must also produce faults that are easy to find and eventually solve.

**Prioritization**

Nice-to-have

**Strategy**

**Example in System**

The login system can be tested and faults can be found easily and corrected if necessary.