**Speed And Quality Of Development**

**Trigger Question:**

How fast must the development be?

How good must the quality of the development be?

**Benefit:**

Higher speed and quality of development will result in finishing the system in time for this subject.

**Impact/Cost:**

Faster development means implementing less functional and nonfunctional requirements, and therefore a less good system.

**Priority:**

1

**Justification:**

Our main goal, first and foremost, is to finish our system to the best standard we can by the end of the subject. Therefore, this is the most important NFR.

**Security**

**Trigger Question:**

Authentication (ascertaining that someone is who they say they are)

Authorization (rules that determine who is allowed to do what)

Do users need to be authenticated?

Do different kinds of users need different authorizations?

**Benefit:**

Users can trust that who they are talking to is who they it says they are, and visa versa.

Administrators can have a higher level of authorization, in order to manage the application.

**Impact/Cost:**

Increased development time and cost.

**Priority:**

2

**Justification:**

If users can not be authenticated then the system will not be useable in the intended manner.

The whole system is based on multiple users talking to each other, they are doing this on the basis that who they are talking to is who it says they are, therefore authentication is very important.

Administrators must be able to manage the system, but regular users must not have this level of control, therefore authorization levels are important.

**Recovery**

**Trigger Question:**

Is there useful information that must not be lost?

**Benefit:**

Information will not be lost, most importantly conversation histories.

**Impact/Cost:**

Increased development time and cost, and possibly the cost of an extra backup database system.

**Priority:**

3

**Justification:**

Users need to be able to trust that their conversations will not be lost. Without this the system is not sufficient.

**Performance**

**Trigger Question:**

Do certain actions need to be performed within a given time period?

**Benefit:**

The app will feel and behave real time, which is its purpose. A chat system that is poor in performance would be frustrating and would not result in a good user experience.

**Impact/Cost:**

Increased development time and cost. And possibly have to keep more information locally to prevent timely lookups.

**Priority:**

4

**Justification:**

For users to have real time conversations, the system needs to feel and behave in real time. Failing to do this would not allow for a useful or enjoyable chat system.

**Usability**

**Trigger Question:**

How usable must the system be?

**Benefit:**

Higher level of user enjoyment and satisfaction, resulting in more customers. Also available to a wider range of people.

**Impact/Cost:**

Increased development time and cost.

**Priority:**

5

**Justification:**

Less people using the system because it is not very usable, is not a good thing. Neither is the people that do use the system not having a pleasant experience.

A high level of usability is important not only for the enjoyment of the customers, but for effective use of the system.

**Reliability and Integrity**

**Trigger Question:**

Can the system afford to be down often and for long periods of time?

**Benefit:**

The less the system is down, the more it can function as it is supposed to. Users may rely on using this system, and it is not acceptable for them to go to use it but be unable to because it is down.

**Impact/Cost:**

Increased development time and cost.

**Priority:**

6

**Justification:**

The system being down regularly or for long periods of time, is not going to make or break the system, but it is something that would best be avoided.

**Capacity**

**Trigger Question:**

What is the throughput need?

What is the storage need?

**Benefit**

The system can function as it is supposed to, without any capacity issues occurring . Also meeting future capacity needs now, will prevent further expansion on the capacity needing to be done, which will save time and money.

**Impact/Cost:**

Higher cost for more expensive hardware.

**Priority:**

7

**Justification:**

Although it is important, our goal is to get the system working. Capacity can always be expanded later on down the line.

**Maintainability**

**Trigger Question:**

Is maintainability important?

**Benefit:**

Can be extended in more flexible ways, making it more likely that future goals will be able to be met. Future changes and maintenance will be cheaper in both time and money.

**Impact/Cost:**

Increased development time and cost.

**Priority:**

8

**Justification:**

The more maintainable the better, but it will not make or break the system.

**Compatibility**

**Trigger Question:**

Is compatibility with different operating systems important?

**Benefit:**

The more operating systems supported, the more people can and will use the system. Also even for people that have one operating system that is supported, it is nice for them to be able to use the system on all of their operating systems.

**Impact/Cost:**

Possibly no to minimal cost to include windows, os x, and linux.  
However other operating systems such as ios and android will result in increased development time and cost.

**Priority:**

9

**Justification:**

Compatibility with extra operating systems would be great, but with our time constraints we are only focussing on desktop environments. However since we are using java, windows, os x, and linux could all be fairly easy to include.