**Non-Functional Requirements Design Procedures**

**Availability**

|  |  |  |
| --- | --- | --- |
| **55** | The new system will be available every day | **High** |
| **56** | The new system will not crash if website traffic increases | **High** |
| **57** | The new system’s website will be available to the public | **High** |
| **58** | The new system will be in place by the end of Spring 2018, unless otherwise stated | **Medium** |

The system will be available every day, will not crash, will be readily available, and will be implemented on time by first using AWS to handle all the server load required to run the system. AWS is very reputable and handles security better than any small server or development team could handle on their end. It also makes sure that these deployable services are always available and ready to use. AWS will ensure that the extra traffic on our system’s deployable services will not create a problem by using queuing if the server load is too heavy that way it will instantiate a service in order instead of trying all at once. We will also host these backend services if needed on a software like docker that is not specific to any OS and will not crash based on a user’s device operating system.

**Storage**

|  |  |  |
| --- | --- | --- |
| **59** | The new system’s data storage will be available to SOS coordinators only | **High** |
| **60** | The new system will provide enough data storage | **High** |
| **61** | The new system will use efficient database design and relationships | **Medium** |
| **62** | The new system’s data will be backed up | **High** |

We will create encryption keys in order to access the AWS’s S3 storage container. Only people with a certain access key will have writing permissions, while other’s only have read permissions, while others may not access it at all. AWS provides system storage that is on a basis where you only pay for what you need. The amount of storage you can upgrade to is virtually infinite and will never be an issue for a company/organization such as SOS. The database will maintain efficiency by eliminating data redundancy and data anomalies by maintaining a strict database design which we will implement, that way data is easily accessible and efficient. The system will also be backed up locally to avoid losing data with a missing internet connection and virtually to avoid loss of the backup incase of device failure.

**Maintenance**

|  |  |  |
| --- | --- | --- |
| **62** | The new system will be easy to maintain | **High** |
| **63** | The new system will be thoroughly tested and not break or have bugs | **High** |
| **64** | The new system maintenance will be available to the IT coordinator | **Medium** |

The system will be updatable using versioning of the system that we can test and manipulate without effecting the current in-use system. Then when we have written enough tests to ensure the update is in fact creating a better system, we can deploy that version to the in-use system. Bugs will be eliminated by testing the system manually and also automatically through the testing framework’s available like Nunit. This will create a system that is easily maintenance and will have a limited amount of bugs.

**Constraints**

|  |  |  |
| --- | --- | --- |
| **65** | The new system will not handle sensitive data about patients | **High** |
| **66** | The new system will not violate HIPAA | **High** |
| **67** | The new system will not interface with the RedCap or Epic Connect patient database | **High** |
| **68** | The new system will allow for disclosure of all volunteers, per the Federal Tort Claims Act | **High** |

The system will only handle data that is specifically needed to create events and pull in volunteers. This way the system doesn’t have sensitive patient data that could be accessed by people that shouldn’t access it. It will not violate HIPAA do to this acceptance of only must-have data. The system will accurately and efficiently store data to allow for full disclosure of volunteer information in case the need arises.