Non-functional Requirements

# 1  Version History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version #** | **Implemented**  **By** | **Revision**  **Date** | **Approved**  **By** | **Approval**  **Date** | **Reason** |
| 1.0 | Andrii Dobroshynskyi | March 20th | Cole Smith | March 21st | Initial Draft |
| 1.1 | Cole Smith | March 22nd | Andrii D. | March 22nd | Refinements |
|  |  |  |  |  |  |

# 1  Key Terms

The following table provides definitions for terms relevant to this document.

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Discord | Messaging service providing socket-based messaging and authentication layers |
| code snippet | A small-sized stand-alone piece of software code that has some output |
| LaTeX | Typesetting language / format for easier rendering of mathematic equations / quantitative concepts |
| P2P | Peer-to-peer connectivity such that communication can be real-time and users can join or leave the service when desired |
| Python | Programming language used for application implementation |

# 2.  Introduction

## 2.1  Purpose Of The Non-Functional Requirements Definition

The Non-Functional Requirements (also referred to as Quality of Service by the International Institute of Business Analysts, Business Analysis Body of Knowledge) Definition documents and tracks the necessary information required to effectively define business and non-functional and technical requirements. The Non-Functional Requirements Definition document is created during the Planning Phase of the project. Its intended audience is the project manager, project team, project sponsor, client/user, and any stakeholder whose input/approval into the requirements definitions process is needed.

Labels:  Definition

## 2.2  Overall Description

The ResearchRealms application is a proposed system that will utilize a socket-based P2P connection to extend functionality of a messaging platform to enhance communication and collaboration between quantitative-focused users. The system will integrate with the Discord messaging and communication platform and build on top of the authentication and messaging layers to provide functionality of executing small-sized snippets of code between connected users, which is beneficial for reproducibility when performing academic research. Additionally, the system will support saving and retrieving the aforementioned code snippets via a specified database connection that can be configured per-user node, and will also support rendering of LaTeX typesetting language for easier viewing of mathematic equations and expressions which is also beneficial when performing quantitative research. The system will also utilize the authentication and messaging layers for these use cases such that outputs of commands can be shared P2P between the nodes in real-time for maximum collaboration.

Labels:  Definition, Analysis

# 3.  Non-Functional Requirements

## 3.1  Hardware Requirements

### 3.1.1  Node server instance

Node hardware running the application and connecting to the dedicated network instance should be able to run the Python language server code that connects to the Discord communication layers

### 3.1.3  Node code execution

Node hardware running the application and connecting to the dedicated network instance should be able to support arbitrary Python language code running

## 3.2  Software Requirements

### 3.2.1  TCP / IP

Software on node running the application and connecting to the dedicated network instance should support the protocol required for socket connections

### 3.2.2  Python support

Software on node running the application should support Python language as well as package installation and execution

### 3.2.3  I / O

Software on node running the application should support basic I / O for code execution output processing and output image rendering

## 3.3  Performance Requirements

### 3.3.1  Code execution timeout

Arbitrary code snippet execution should complete within the 30 second timeout limit

Labels:  Performance Requirement

### 3.3.2  Database reads

Accesses to the database via the DB connection for snippet persistence and retrieval should not exceed 1s

Labels:  Performance Requirement

### 3.3.3  File writes

Writes to disk and reads for output rendering should not exceed 1s

Labels:  Performance Requirement

### 3.4.2  P2P connectivity limits

Application should support < 1000 connected users to a dedicated network of nodes for communication without difference in performance

Labels:  Performance Requirement

## 3.4  Supportability Requirements

### 1  Database configuration

Database connection string should be configurable without difference in performance

### 3.4.2  Command testability

Application software should be testable via a specified command to test command validity

### 3.4.3  Connection testability

Application software should be testable via a specified command to test connectivity to Discord communication layer as well as database connection

## 3.5  Security Requirements

### 3.5.1  User identification

Software should connect to the authentication layer and the context layer to identify users

### 3.5.2  Code execution restriction

Software should restrict code execution that attempts to use system resources

### 3.5.3  Code execution clean-up

Software should remove written files on disk after code execution completion

## 3.6  Interface Requirements

### 3.6.1  Text output

Software should connect to the messaging layer to send text data via a dedicated socket

### 3.6.2  Image output

Software should connect to the messaging layer to serialize image byte data and send via a dedicated socket

### 3.6.3  Text input

Software should connect to the messaging layer to accept incoming data streams via connections to peers

## 3.7  Availability Requirements

### 3.7.1  Fault auto restart

Application should automatically restart after an exception

### 3.7.2  Fault downtime threshold

Application should maintain at least 2 nines (99.0%) availability threshold equivalent to allowing 3 days, 15 hours and 40 minutes downtime per year

## 3.8  Assumptions / Constraints

### 3.8.1  Network connectivity

Application assumes available connection to the internet

### 3.8.2  TLS / SSL certificates

Application assumes support for SSL / TLS for secure communication and access to the authentication layer

# Appendix A:  Non-Functional Requirements Definition Approval

The undersigned acknowledge they have reviewed the **Non-Functional Requirements Definition** and agree with the approach it presents. Any changes to this Requirements Definition will be coordinated with and approved by the undersigned or their designated representatives.

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Date: | March 20th, 2020 |
| Print Name: | Andrii Dobroshynskyi |  |  |
| Title: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |  |
| Role: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |  |
| Signature: | *Cole Smith* | Date: | March 20th, 2020 |
| Print Name: | Cole Smith |  |  |
| Title: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |  |
| Role: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |  |