Cryptocurrency Market Simulator

Supplementary Specification

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <21/03/18> | <1.0> | Supplementary Specifications of the Cryptocurrency Market Simulator | Barabas Hunor |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

2. Non-functional Requirements 4

2.1 Availability 4

2.2 Performance 4

2.3 Security 4

2.4 Testability 4

2.5 Usability 4

3. Design Constraints 4

Supplementary Specification

# Introduction

This document includes supplementary information that was not specified elsewhere, but is still important to the project. These include:

-application standards

-legal and regulatory requirements

-requirements for attributes of availability, performance, security, testability and usability

- design constraints

# Non-functional Requirements

## Availability

* **Quality attribute definition:** Measure of application uptime ( the extent to which the application provides the required functionality)
* **Source of stimulus:** The end user
* **Stimulus:** Launching the application
* **Environment:** idle
* **Artifact:** the whole system
* **Response:** availability to perform functionalities
* **Response measure:** the time it takes for the application to be ready to provide functionalities
* **Tactics:** Having a clean architecture and proper security to prevent downtime

## Performance

* **Quality attribute definition:** The speed at which the application performs the requested functionalities
* **Source of stimulus:** The End User
* **Stimulus:** Initiating any functionality of the application, for example a buy order
* **Environment:** The system is idle, ready to accept instructions
* **Artifact:** Depends on the requested functionality, usually involves all layers
* **Response:** the requested functionality is performed
* **Response measure:** amount of time passed between request and delivery of result
* **Tactics:** performance tuning

## Security

* **Quality attribute definition:** The system’s preparation for malicious attacks
* **Source of stimulus:** Developer
* **Stimulus:** Making an artificial malicious attack
* **Environment:** Application under development
* **Artifact:** Any layer that needs to be seucured
* **Response:** attack failed
* **Response measure:** How well the system is prepared for various attacks
* **Tactics:** using basic security principles like encapsulation plus more advanced measures if needed

## Testability

* **Quality attribute definition:** Measure of how different components of our application can be isolated and tested to see if they provide their functionality as intended
* **Source of stimulus:** Developer
* **Stimulus:** Performing isolated tests
* **Environment:** under development ( or even after)
* **Artifact:** one part of the system ( can be from any layer)
* **Response:** performed test
* **Response measure:** test result
* **Tactics:** Taking testability into consideration right from the beginning

## Usability

* **Quality attribute definition:** Measure of how well the system meets the user’s requirements. This involves everything that contributes to the user experience.
* **Source of stimulus:** End User
* **Stimulus:** using the application
* **Environment:** finished application
* **Artifact:** Presentation Layer
* **Response:** User is getting the expected responses to his/her actions
* **Response measure:** smooth, successful completion of tasks
* **Tactics:** put ourselves in the user’s perspective and develop accordingly

# Design Constraints

The application is developed using the Java programming language. The platform is desktop based. An internet connection is required to provide live prices of currencies.