



# Software Requirement Specification

## Version 0.5

### Project Supervisor :

- Dr. Sanjeewa Weerawarana

### Super Group Head :

- Mr. Indika Perera

### Group Members :

- Baranasuriya N. T. (050043D)
- Boralugoda M. B. R. C. (050048X)
- Nafran M. S. M. A. (050285U)

Project Supervisors:

Dr. Sanjeewa Weerawarana

Mr. Indika Perera

Super Group Head:

Mr. Indika Perera

Group Members:

Baranasuriya N. T. (050043D)

Boralugoda M. B. R. C. (050048X)

Nafran M.S.M.A. (050285U)

## Contents

1	Introduction	4
1.1	Purpose.....	4
1.2	Target Audience .....	4
1.3	Project Scope.....	4
2	Overall Description	6
2.1	Product Perspective .....	6
2.2	Product Features .....	6
2.3	User Characteristics.....	6
2.4	Assumptions and Dependencies .....	7
2.5	Operating Environment.....	8
2.6	Design and Implementation Constraints.....	8
2.7	User Documentation .....	8
3	System Features	9
3.1	Enabling the Service Runtime.....	9
3.2	Disabling the Service Runtime.....	12
3.3	Registering the Service Runtime in the MOINC Server .....	14
3.4	Accepting Tasks from the MOINC Server .....	16
3.5	Executing Received Services.....	18
3.6	Stopping Execution of Received Services .....	19
3.7	Un-registering the Service Runtime from the MOINC Server .....	21
3.8	Calculating Points Based on Time Spent for Execution.....	22
4	External Interface Requirements	23
4.1	User Interfaces .....	23
4.2	Communication Interfaces .....	23
5	Non functional Requirements	24
5.1	Performance Requirements .....	24
5.2	Input/Output Requirements .....	24
5.3	Safety Requirements .....	24
5.4	Software Quality Attributes .....	24
6	Abbreviations	26

# 1 Introduction

## 1.1 Purpose

The purpose of this document is to state and describe the software requirements for the MOINC (Moratuwa Open Infrastructure for Network Computing) client module. This software development project is carried out with the main intention of fulfilling the requirements of the CS-4200 module conducted by the Department of Computer Science and Engineering, University of Moratuwa under the super group of Software Architecture. This document basically covers all the major functional, non-functional, performance, security and safety requirements of the MOINC client. This document also provides the basis for scope, design and development of the entire project and it will constitute all the planning, estimation and development activities associated with the project.

## 1.2 Target Audience

This requirements specification document is deliberated to be a reference material for all the academic staff members of the Department of Computer Science and Engineering, University of Moratuwa who are involved in supervising and evaluating projects carried out under the module CS-4200. Also this is targeted towards all the mentors and industrial supervisors associated with the MOINC client module. Furthermore this document will function as a primary reference material for all the developers engaged in this project and the projects associated.

## 1.3 Project Scope

MOINC is a project which enables the users of the idle computers to voluntarily join the MOINC network and let it use the processing power of their computers when the resources remain idle and/or when the user wants, which is beneficial for the users in financial terms as well.

In addition of fulfilling the main objectives of project MOINC, MOINC client module is mainly responsible in monitoring the local system activities. Notifying the MOINC server when the client becomes idle and when it requires the processing time, could be considered as the main duties of the client module. Other than that it should provide room for a SOAP execution environment which takes care of executing the services provided by the server. Calculation of points based on how much processing power was used

and sending information about these points, service execution history, etc. to the server are also the client's responsibilities. Also the client module should be developed with the ability of installable in multiple platforms.

When developing the system we will only target the features and the services mentioned in this specification document. The client shall not get involved in the tasks managed by the server module. The client will work with a set of protocols that are agreed between itself and the server.

## 1.4 Document Conventions

Font:

Heading Type	Font Size (pt)	Font Type	Font Colour
Major Headings	14	Cambria (Headings)	Blue, Accent 1, Darker 25%
Sub Headings	13	Cambria (Headings)	Blue, Accent 1
Other Headings	11	Cambria (Headings)	Blue, Accent 1
Body Text	11	Calibri (Body)	Black

## 2

## 2 Overall Description

### 2.1 *Product Perspective*

MOINC service runtime will be a standalone client application. It will be installed in all the client machines which will agree on working inside the volunteer computer network which will execute web services when the PC is idle.

### 2.2 *Product Features*

MOINC client application will convert any PC to a java Web Service execution Platform. Following are the key features of the MOINC client application. These features are described in detail later in this document.

- Run as a background program in the PC
- Enable or Disable the Service Runtime software
- System resources which are used to be configurable by the end user
- Work as a screen saver

### 2.3 *User Characteristics*

MOINC client application user can be a worker who will be working in an organization which will be using MOINC to get there organizational service requirements executed using the idle machines. Otherwise the user can be a common internet user who will be willing to contribute to the MOINC network so that he/she will gain something in return.

Therefore the MOINC client application will be a simple to install and ready to use kind of software which will not require any high level IT knowledge.

## ***2.4 Assumptions and Dependencies***

MOINC client application will be using the JVM to execute services within an OS environment. The client application will rely heavily on the services provided by the MOINC server and the server management components.

## ***2.5 Operating Environment***

### **2.5.1 Hardware Requirements**

The client module will at least need a computer having the following configuration to run smoothly.

- Intel Pentium 3, 1.0 GHz or equivalent are above
- System memory 256 MB or above.
- The ability to display 8-bit graphics at 800x600 resolution
- Network connectivity
- 200 MB Hard Disk Drive with at least 50 MB available for use

### **2.5.2 Software Requirements**

The following pre-requisites are needed in the local computer that the client application will run in.

- JVM
- A web service execution environment.

## ***2.6 Design and Implementation Constraints***

It is expected that the MONIC client application will work without any user interactions once the installation and the basic configuration is done (MOINC client will have some per-defined configurations). At the same time the MONIC client application is expected to inform the server whenever there is a problem in executing the given service within the given time line.

Since the project development time period is fixed and cannot be extended MONIC client application initially will be focused on the main requirements and features.

## ***2.7 User Documentation***

A comprehensive user manual will be given with each release of the client application. The user manual will be of HTML format and will be completely read via the internet. This manual will describe all the important aspects of the MOINC application and will include step-by-step guidelines for installing and configuring the application.

In addition, since the application will be developed and distributed under the open source principles the full source code, documentation and other design documents will be made available in the project website. Also a public mailing list will be opened up for user queries and all the mail archives will be made public.

Currently all the discussions related to the project take place in a private mailing list. Mail archives related to these discussions will be also made public in the future.

## 3 System Features

### 3.1 Enabling the Client Application

#### 3.1.1 Description

The client application that will be installed on the volunteering computers will run when the system becomes idle. This client will need to be enabled for it to run when the system becomes idle. Thus enabling of the system is needed for it to receive tasks to perform from the central server.

#### 3.1.2 Priority

Priority – High

#### 3.1.3 Use Cases

Use Case #/ID	C1	Stage	Configuration Management
Use Case Overview			
Turn on the client application on the local machine			
Pre-Conditions			
<ul style="list-style-type: none"> <li>The local machine should be connected to the network</li> <li>The application should be installed on the machine</li> </ul>			
Main List of Events/Flow of Events			
<ul style="list-style-type: none"> <li>The user opens the configuration view</li> <li>Selects enable option</li> </ul>			

Terminating Outcome	Condition Affecting Terminating Outcome	Post-Conditions
The system will be successfully configured to access the central server	Successful installation and start-up of the system	The application will communicate with the server when the machine is idle

Use Case #/ID	C2	Stage	Configuration Management
Use Case Overview			
Monitoring the local computer			
Pre-Conditions			
<ul style="list-style-type: none"> <li>The local machine should be connected to the network</li> <li>The system should be installed and enabled on the machine</li> </ul>			
Main List of Events/Flow of Events			
<ul style="list-style-type: none"> <li>The system will work in the background. No particular events are necessary</li> </ul>			
Terminating Outcome	Condition Affecting Terminating Outcome	Post-Conditions	
The application will start communicating with the server in order to perform some executions	The machine becomes idle; hence the processing power can be used by MOINC.	The client will start executing the services it received from the server	
The application will continue to wait to contact the server	The machine is not idle. Its processing power is used by the user for some activity(s).	The application will continue to run in the background without contacting the server.	

### 3.2 *Disabling the Client Application*

#### 3.2.1 Description

The client application should give the user the option of switching it completely off as well. In other words the system should accept users who do not want to give up their processing power even when the machine is idling. To fulfil this requirement, it is essential to have this feature incorporated to the client.

#### 3.2.2 Priority

Priority – High

#### 3.2.3 Use Cases

Use Case #/ID	C3	Stage	Configuration Management
Use Case Overview			
Turn off the client application on the local machine			
Pre-Conditions			
<ul style="list-style-type: none"> <li>The local machine should be connected to the network</li> <li>The client should be installed on the machine</li> </ul>			
Main List of Events/Flow of Events			
<ul style="list-style-type: none"> <li>The user opens the configuration view</li> <li>Selects disable option</li> </ul>			
Terminating Outcome	Condition Affecting Terminating Outcome	Post-Conditions	
The client will be successfully configured not to access the central server even if the machine is idle	Successful installation and start-up of the system	The application will not communicate with the server when the machine is idling	



### 3.3 Registering the Client Application in the MOINC Server

#### 3.3.1 Description

Once the system is installed in the local machine the server should be notified about the newly added node to the cluster. Hence there should be a mechanism for the system to inform the server about the availability of the new node. This is done through the register feature found in the system. Once the user registers the newly installed system with the central server, it will get tasks to perform.

#### 3.3.2 Priority

Priority – High

#### 3.3.3 Use Cases

Use Case #/ID	C4	Stage	Configuration Management
Use Case Overview			
Register the client with the central MOINC server			
Pre-Conditions			
<ul style="list-style-type: none"> <li>The local machine should be connected to the network</li> <li>The system should be installed and enabled on the machine</li> <li>The MOINC server should be up and running</li> </ul>			
Main List of Events/Flow of Events			
<ul style="list-style-type: none"> <li>The user opens the configuration view</li> <li>Selects Register option</li> </ul>			
Terminating Outcome	Condition Affecting Terminating Outcome	Post-Conditions	
The system will be successfully registered with the server	Successful communication between the server and the system via a well defined set of protocols	The client will get tasks to perform when it's idling.	



### 3.4 Accepting Tasks from the MOINC Server

#### 3.4.1 Description

Once the system is installed, enabled and registered in the server it is ready to accept services to be executed from the server. As soon as the local machine goes into idling mode the system will fire up and contact the MOINC server to notify that now it's free. When the server receives this notification it will hand down a service to be executed to the client.

#### 3.4.2 Priority

Priority – Very High

#### 3.4.3 Use Cases

Use Case #/ID	C5	Stage	Service Receiving
Use Case Overview			
Receiving services from the MOINC server			
Pre-Conditions			
<ul style="list-style-type: none"> <li>The local machine should be connected to the network</li> <li>The application should be installed and enabled on the machine</li> <li>The MOINC server should be up and running</li> </ul>			
Main List of Events/Flow of Events			
<ul style="list-style-type: none"> <li>The local machine goes into the idling state</li> </ul>			
Terminating Outcome	Condition Affecting Terminating Outcome		Post-Conditions
The application will request services from the server	Successful communication between the MOINC server and client		The application will start executing the service



## 3.5 Executing Received Services

### 3.5.1 Description

After receiving a service from the MOINC server, the service runtime will work on executing it.

### 3.5.2 Priority

Priority – Very High

### 3.5.3 Use Cases

Use Case #/ID	C6	Stage	Service Execution
Use Case Overview			
Executing services sent from the MOINC server			
Pre-Conditions			
<ul style="list-style-type: none"> <li>The local machine should be connected to the network</li> <li>The system should be installed and enabled on the machine</li> <li>The MOINC server should be up and running</li> </ul>			
Main List of Events/Flow of Events			
<ul style="list-style-type: none"> <li>The local machine goes into the idling state</li> </ul>			
Terminating Outcome	Condition Affecting Terminating Outcome	Post-Conditions	
The runtime will successfully execute the service handed down by the server	Successful communication between the MOINC server and application	Successful execution of the received service	

## 3.6 Stopping Execution of Received Services

### 3.6.1 Description

As mentioned in the proposal, the service runtime will execute services only when the local machine is idling. Therefore once the machine comes back online service execution will be stopped.

### 3.6.2 Priority

Priority – Very High

### 3.6.3 Use Cases

Use Case #/ID	C7	Stage	Service Execution
Use Case Overview			
Stopping execution of services sent from the MOINC server			
Pre-Conditions			
<ul style="list-style-type: none"> <li>The local machine should be connected to the network</li> <li>The system should be installed and enabled on the machine</li> <li>The MOINC server should be up and running</li> </ul>			
Main List of Events/Flow of Events			
<ul style="list-style-type: none"> <li>The local machine goes into the idling state</li> <li>The service runtime starts service execution</li> <li>The local machine goes into a non idling mode</li> </ul>			
Terminating Outcome	Condition Affecting Terminating Outcome	Post-Conditions	
The runtime will stop execution and gives the processing power to the local machine	The processing power is required by the local machine	Execution will be stopped by the service runtime  The runtime will then inform the server that execution was	

		stopped
--	--	---------

## 3.7 Un-registering the Client from the MOINC Server

### 3.7.1 Description

The user should have the opportunity of un-registering the client from the MOINC server. When un-registered the system will not be handed any services for execution from the server.

### 3.7.2 Priority

Priority - High

### 3.7.3 Use Cases

Use Case #/ID	C8	Stage	Configuration Management
Use Case Overview			
Un-register the client application from the MOINC server			
Pre-Conditions			
<ul style="list-style-type: none"> <li>The local machine should be connected to the network</li> <li>The service runtime should be installed and enabled on the machine</li> <li>The MOINC server should be up and running</li> </ul>			
Main List of Events/Flow of Events			
<ul style="list-style-type: none"> <li>The user opens the configuration view</li> <li>Selects Un-Register option</li> </ul>			
Terminating Outcome	Condition Affecting Terminating Outcome	Post-Conditions	
The runtime will be successfully un-registered with the server	Successful communication between the server and service runtime via a well defined set of protocols	The runtime will not get services for execution hereafter	

### 3.8 Calculating Points Based on Time Spent for Execution

#### 3.8.1 Description

The service runtime will receive points based on the time it spends on executing services handed by the server. These points will be used by the server management console to rank the registered users of the network.

#### 3.8.2 Priority

Priority – Low

#### 3.8.3 Use Cases

Use Case #/ID	C10	Stage	Points Calculation
Use Case Overview			
Calculate points based on the processing time required by the services			
Pre-Conditions			
<ul style="list-style-type: none"> <li>The local machine should be connected to the network</li> <li>The service runtime should be installed and enabled on the machine</li> <li>The MOINC server should be up and running</li> <li>The service runtime should be executing services pushed by the server</li> </ul>			
Main List of Events/Flow of Events			
<ul style="list-style-type: none"> <li>The server hands down a service to the service runtime</li> <li>Runtime starts executing the service</li> </ul>			
Terminating Outcome	Condition Affecting Terminating Outcome	Post-Conditions	
The runtime will calculate points based on the time spent on executing the services	Execution of the handed services	Inform the server about the gathered points	

## 4 External Interface Requirements

### 4.1 User Interfaces

As this application will be used by all types of users including ones who have no technical expertise. The user interfaces will be designed in such a way that all types of users will find it easy to use and manage. In other words the interfaces should be quite user friendly. The GUIs should give the following features to the user.

- An interface to configure the client (which includes features such as registering the client in the server etc)
- An interface which shows the statistical information about the services that were executed

### 4.2 Communication Interfaces

As mentioned earlier, the client application relies heavily on the services granted by the MOINC server. Hence the client should deal with well defined APIs of the server to perform its activities. The communication should take place over protocols like HTTP. Hence the communication interfaces should support protocols such as HTTP as well as the protocols that we come up during the development life cycle of the project.

## 5 Non functional Requirements

### 5.1 Performance Requirements

The main feature of the client application is executing the received services in an efficient manner. As the user may allow the application to fully utilize the processing power of the machine, the application should perform in such a way that the processing power is used optimally. Hence the client application will follow service execution standards which will achieve the above mentioned objective.

When engaged in service execution, the client application will make sure that the response time is kept to a minimum always. Otherwise the users of the system will get tired of using the MOINC product suit.

### 5.2 Input/Output Requirements

The client application will not be allowed to access the entire file system of the local machine when it's in operation. A working directory will be given to the application and it will have access to that particular directory only.

When the client application wants to communicate with the server and the server enhancement modules it will use standard data representation formats such as XML and will operate over standard protocols such as HTTP.

### 5.3 Safety Requirements

The client application will not be allowed to make any system calls or run any malicious scripts while in operation. As the system will be built on top of the JVM additional security is provided as well.

The web service execution environment will not have direct access to the local machine Therefore the local machine will be protected from harmful operations that might take place.

### 5.4 Software Quality Attributes

#### Reliability

The client application will run only when the system is idle. Therefore as soon as the machine goes into the non idling state complete processing power should be granted to the user, thus stopping all service executions.

If the user has setup some policies (e.g. allowing the application to use only 50% of the processing power etc) the client application should respect these policies and working according to them. In other words the system should work as the user expects and not perform any surprising tasks.

### **Safety**

The client application will not be allowed direct access to the local machine to prevent harmful operations from taking place.

### **Security**

The client application will have access only to a single working directory in the file system thus disabling unauthorized access to the other directory structures of the system.

## 6 Abbreviations

Abbreviation	Meaning
<b>MOINC</b>	Mora Open Infrastructure for Network Computing
<b>JVM</b>	Java Virtual Machine
<b>SOAP</b>	Simple Object Access Protocol
<b>PC</b>	Personal Computer
<b>HTML</b>	Hyper Text Markup Language