Software Requirements Specification

for

Visualization Tool for Composition of Cloud Computing Services

Version v0.0.0.1

Prepared by Serrano, Gonzalo Ramirez, Zolangi Baral, Sudip Troung, Johnson Choi, Hongsuk

NASA DIRECT-STEM/Faculty

December 8, 2017
Table of Contents

1. Introduction .................................................................<pg 4>
   1.1. Purpose .................................................................<pg 4>
   1.2. Intended Audience and Reading Suggestions .........<pg 4>
   1.3. Product Scope .......................................................<pg 4>
   1.4. Definitions, Acronyms, and Abbreviations ..........<pg 5>
   1.5. References .........................................................<pg 5>
2. Overall Description .....................................................<pg 6>
   2.1. Product Perspective .............................................<pg 6>
   2.2. Product Functions ...............................................<pg 7>
   2.3. User Classes and Characteristics ......................<pg 7>
   2.4. Operating Environment ......................................<pg 8>
   2.5. Design and Implementation Constraints .............<pg 8>
   2.6. User Documentation ............................................<pg 8>
   2.7. Assumptions and Dependencies .........................<pg 8>
   2.8. Apportioning of Requirements .............................<pg 8>
3. External Interface Requirements ..................................<pg 9>
   3.1. User Interfaces ...................................................<pg 9>
   3.2. Hardware Interfaces ..........................................<pg 13>
   3.3. Software Interfaces ...........................................<pg 13>
   3.4. Communications Interfaces .................................<pg 13>
4. Requirements Specification ........................................<pg 14>
   4.1. Functional Requirements ....................................<pg 14>
   4.2. External Interface Requirements ........................<pg 18>
   4.3. Logical Database Requirements ...........................<pg 18>
   4.4. Design Constraints ............................................<pg 19>
5. Other Nonfunctional Requirements .............................<pg 20>
   5.1. Performance Requirements .................................<pg 20>
   5.2. Safety Requirements ..........................................<pg 20>
   5.3. Security Requirements .......................................<pg 20>
   5.4. Software Quality Attributes ...............................<pg 20>
   5.5. Business Rules ..................................................<pg 20>
6. Other Requirements ..................................................<pg 21>
Appendix A: Glossary ......................................................<pg 22>
Appendix B: Analysis Models .........................................<pg 23>
## Revision History

<table>
<thead>
<tr>
<th>Name</th>
<th>Date</th>
<th>Reason For Changes</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draft #1</td>
<td>12/8/2017</td>
<td>First draft</td>
<td>v0.0.0.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. **Introduction**

This section gives a scope description and overview of everything included in this SRS document. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.

1.1 **Purpose**

The purpose of this document is to give a detailed description of the requirements for the “Visualization Tool for Composition of Cloud Computing Services” (VTCCCS) software. It will illustrate the purpose and complete declaration for the development of the system. It will also explain system constraints, interface and interactions with other external applications.

1.2 **Intended Audience and Reading Suggestions**

This document is intended for all individuals participating in and/or supervising the VTCCCS project. Readers interested in a brief overview of the VTCCCS software should focus on part 1 (Introduction), as well as Part 2 of the document (Overall Description). These sections provide a brief overview of each aspect of the project as a whole.

Part 3 (External Interface Requirements) offers further technical details, including information on the user interface as well as the hardware and software platforms on which the application shall run.

Readers who have not found the information they are looking for should check Part 6 (Other Requirements), which includes any additional information which does not fit logically into the other sections.

1.3 **Product Scope**

The VTCCCS system is composed of two main components: a client-side application which will be served over the web, and a server-side which will support and interact with various client-side features. The system is designed to facilitate the process of tracking and displaying the composition of cloud services and the results of these composite services. Potential scenarios
include processing satellite data and then passing the processed data to another service to analyze.

1.4 Definitions, Acronyms, and Abbreviations

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTCCCS</td>
<td>Visualization Tool for Composition of Cloud Computing Services</td>
</tr>
<tr>
<td>User</td>
<td>Someone who interacts with the client-side application</td>
</tr>
<tr>
<td>Admin/Administrator</td>
<td>System administrator who is given specific permission for managing and controlling the system</td>
</tr>
<tr>
<td>Node</td>
<td>A service generated by the user that will send or receive information from an external/internal web service</td>
</tr>
<tr>
<td>View Block</td>
<td>This is where nodes will be displayed depending on the View selected</td>
</tr>
<tr>
<td>View</td>
<td>Determines how nodes shall be displayed to the user</td>
</tr>
<tr>
<td>Red/Cyan</td>
<td>The state of node when toggled</td>
</tr>
<tr>
<td>Tool</td>
<td>Refers to VTCCCS</td>
</tr>
<tr>
<td>REF</td>
<td>Reference to something, etc. Eg.) Functional Requirement, FR</td>
</tr>
</tbody>
</table>

1.5 References

All references used in the creation of this document are listed below.

- The Java EE 5 Tutorial, “Creating a Simple Web Service and Client with JAX-WS”, Oracle, September, 2010
- The Java EE 6 Tutorial, “Types of Web Services”, Oracle, January 2013
2. Overall Description

This section will give an overview of the visualization tool. The functions and components will be explained in its context to show how it works. It will also describe the functionalities available to users and admins. Finally, assumptions and constraints will be presented.

2.1 Product Perspective

The Visualization Tool will have two types of web portals. One for users to create projects and view relevant information of web services while the other is for admins to manage user information and other details.

There will be a number of functionalities available to the user to create projects to simulate cloud computing. The Tool will need to attempt to connect to a web service of the URL that the user would have provided in a project.

A database will be used to store information and take information from, namely user credentials and their projects.

Admins will be met with a different set of functionalities to manage user data of which will also modify data in the database.
2.2 Product Functions

Users will be able to create, save, load, delete, and modify projects. These projects are used to simulate cloud computing and views its QoS.

Once a user chooses to load or create a new project, they will be taken to the View Block. Nodes can be created and displayed in the View Block. The View Block displays the nodes depending on the View selected of which the user can choose based on their preference.

The first View displays the children nodes of a parent node. The root node is displayed by default. In this View, users can edit their projects. The user shall be able to right click an empty spot in the View Block to bring up a number of options, such as creating nodes, or right click on an existing node for other options.

When creating or modifying a node, the user will be prompted to enter in relevant information in the appropriate fields, such as node title.

Admins will have a number of functionalities to manage user data.

2.3 User Classes and Characteristics

There are two types of users that will interact with the system: Users that will use the tool and Admins that will manage user data,

The Users will use the major functions of the Tool, which is to create projects to simulate cloud computing. This means the user should be capable of navigating around the web portal and manage their projects as well as being able to execute the project to retrieve relevant information.

Admins will not be able to modify projects or view them. They are only able to manage user data.
2.4 Operating Environment

This software shall operate on any computer that has a modern web browser that supports Javascript.

2.5 Design and Implementation Constraints

The Internet is a constant constraint as it needs to be connected to use web services and to communicate with the server side functions. This is crucial as it is the only way to accurately simulate cloud computing as well as store and load information from the database.

External web service API can be a constraint since we need to know what data we will need to parse out of the service results.

2.6 User Documentation

No user Documentation has been written at this time.

2.7 Assumptions and Dependencies

Google Gson is a dependency to convert a class into a JSON message to the RESTful functions. If this library were to declare deprecated functions we would need to update the backend functions.

2.8 Apportioning of Requirements

If there any requirements to be added in later, it shall be added to the 4th section in the next revision of this document.
3. **External Interface Requirements**

This section will provide a detailed description of inputs and outputs of the Tool. Prototype user interface will be described as well as hardware, software, and communication interfaces if needed.

3.1 **User Interfaces**

At the web portal, the user will be met with a home page where the user may login.

*Figure 1 - Home Page*
If the user is a first time user, then they may register for an account. If the user is returning user or has just registered, they may login.

![Login/Register](image)

*Figure 2 - Login/Register*

Once user has logged in, they will be taken to a project list page where all their saved projects will be listed. It will be initially empty if user is newly registered. User may choose to create new projects, edit existing projects, and view projects.

![Project List](image)

*Figure 3 - Project List (Note, only shows list of nodes. Will update in later revision)*
This is the Editor View in the View Block. Here users may edit their projects by creating, modifying, and deleting nodes. The root node will be displayed by default with its children, or none if it is a new project. The user may right click on the View Block to list out a number of options to choose from.

1. **Insert:** Creates a new node in the View Block
2. **Delete:** Deletes selected node
3. **Edit:** Edits selected node
4. **Composite:** Composites Red nodes
5. **Composition:** Will enter in a node to view children nodes if any.
6. **Performance:** A window will popup detailing QoS information of the node
7. **Back:** Backs out from within a parent node

*Figure 4 - Editor View, Right Click Options*
If a user chooses to edit a node, they will be taken to this page where they may modify node details and add services.

1. **ID**: The ID of the node, cannot be edited.
2. **URL**: The URL of the webservice
3. **TITLE**: Name of node
4. **DESCRIPTION**: A text block describing the node
5. **PARAMETER**: Inputs the nodes will receive

![Figure 6 - Node Edit](image)

Users are able to modify nodes to their preference and how the execution of the simulation would proceed. Users can drag and hold nodes to change its position. If holding shift, users can draw arrows from a node to another node, to indicate that node is outputting to the node the arrow points to. When a node is selected, it will have a blue ring, indicating that it is selected, and is able to be deleted or modified. Nodes can also be clicked upon to toggle its state, from Cyan to Red and vice versa. Red nodes are subjected to a composite, which will automatically create a parent node for the Red nodes as seen in the picture. All parent nodes, auto generated or created by the user, will have a number underneath its title indicating how much children it currently holds. It does not count grandchildren or children beyond. Users can also enter into a node by selecting Composition and the View Block will repopulate with the children of the new parent node or it will be empty if parent node does not have children.
3.2 Hardware Interfaces

The Tool does not interact with external hardware.

3.3 Software Interfaces

- The Tool requires the latest MySQL version
- Tool requires the use of RESTful web services to communicate with web services in the cloud as well as communicate with the back-end.
- The VTCCCS shall interact with outside services through a JSON formatted message.

3.4 Communications Interfaces

- The Tool will communicate with the web services in the cloud and the server side database via RESTful web services XML.

*Figure 7 - Editor View, Node States*
4. Requirements Specification

This section contains all of the functional and quality requirements of the system. It gives a detailed description of the system and all its features.

4.1 Functional Requirements

- **REF: FR1**
  - User Registration
  - User must register an account if this is their first time using the Tool.
    - User shall provide username, password, and email address.
  - Depends on: FR1

- **REF: FR2**
  - User Login
  - User should have registered for an account and will be able to provide their credentials when choosing to login.
  - Depends on: FR1

- **REF: FR3**
  - Project List
  - A project list shall load in saved projects from the database with options.
  - Depends on: FR2

- **REF: FR4**
  - Create Project
  - A new project shall be created for the user to modify and use.
  - Depends on: None

- **REF: FR5**
  - To Graph
  - This shall bring the user to the View Block to graphically display their project and allowing the user to edit it.
  - Depends on: FR3
• **REF: FR6**
  ○ **View**
  ○ This shall determine how nodes are displayed in the View Block.
  ○ Depends on: None

• **REF: FR7**
  ○ **View Block**
  ○ This shall display the project graphically as nodes and open up a number of options for the user to edit the nodes.
  ○ Depends on: FR6

• **REF: FR8**
  ○ **Left Click Select**
  ○ When a user clicks on a node, the node shall elicit a blue ring around the node.
  ○ Depends on: FR7

• **REF: FR9**
  ○ **Toggle Node State**
  ○ The node shall change state, indicated by color, between Cyan and Red.
  ○ Depends on: FR7, FR8

• **REF: FR10**
  ○ **Draw Arrow**
  ○ An arrow shall be drawn based on the mouse cursor location.
  ○ Depends on: FR7, FR8

• **REF: FR11**
  ○ **Arrow Connect**
  ○ Arrow shall connect the starting node to the ending node.
  ○ Depends on: FR10
- **REF: FR12**
  - Input To Output
  - This shall take inputs from the starting node, the start of the arrow, and outputs them to the ending node, the end of the arrow.
  - Depends on: FR10

- **REF: FR13**
  - Right Click Display
  - A small window shall appear near the mouse cursor with a list of edit options for the user.
  - Depends on: FR7

- **REF: FR14**
  - Insert
  - A new node shall be created on the View Block.
  - Depends on: FR7, FR13

- **REF: FR15**
  - Delete
  - Selected node shall be deleted.
  - Depends on: FR7, FR8, FR13

- **REF: FR16**
  - Edit
  - This shall bring the user to a different page to edit a selected node.
  - Depends on: FR8, FR13

- **REF: FR17**
  - Composite
  - This shall composite all Red nodes and auto generate a parent node to hold them.
  - Depends on: FR9, FR13
- **REF:** *FR18*
  - **Composition**
  - This shall allow the user to enter within a selected node and display its children nodes within if any.
  - Depends on: FR8, FR7, FR13

- **REF:** *FR19*
  - **Performance**
  - A window popup shall appear to detail the QoS data of the selected node to the user.
  - Depends on: FR13, FR8

- **REF:** *FR20*
  - **Back**
  - This shall bring the user one level higher in the node tree, backing out of a parent node and displaying the set of nodes of the level. It shall not do anything if the next node level up is the root node.
  - Depends on: FR13, FR7

- **REF:** *FR21*
  - **Edit Node**
  - This shall display fields for the user to provide new or modify details of the node including web service url, node title, description, and parameters.
  - Depends on: FR16

- **REF:** *FR22*
  - **Add Parameter**
  - This shall add a parameter to be used for the node. The user may then input information.
  - Depends on: FR21
4.2 External Interface Requirements

4.2.1 Users
The user shall input their First Name, Last Name, Email and Password to be stored in the Database in the Users table.

Users of the tool shall use this information to access persistent account information from the Database.

4.2.2 Projects
Users of the tool shall create Projects with inputs that include Name, nodes, and services.

Projects shall be stored as persistent account bound information on the Database.

4.2.3 Services
Users shall input Parameters, URL, Name, and Output container for each service node.

The VTCCCS shall use the URL input to execute the external service.

The node shall parse the result of the service for the input parameters required for the tool to execute.

The node shall share the results of the parsed data with a different node that requires the parsed data.

4.3 Logical Database Requirements

- **USERS**: The user shall store their account information on the database.
- **SERVICE**: Users shall store URL addresses to external Services on the database.
● **PROJECTS**: Users shall store project hierarchies on the database.
● **NODE**: Users shall store nodes which represent services on the database.
● **INPUT**: Users shall store Inputs required by services on the database.
● **OUTPUT**: Users shall store Outputs required by the tool on the database.
● **XRefUsersProjects**: Users shall store projects associated with their account on the database.
● **RefServiceXNode**: Nodes shall be stored with information about the service they represent on the database.
● **RefNeighbors**: Nodes shall be stored with information about other nodes it interacts with.
● **RefChildrenNodes**: Nodes shall be stored with information required by other nodes.

### 4.4 Design Constraints

Since we are depending on third party web services a constraint is the result these external services render. External services have different ways of presenting information and a potential problem will be extracting only the information crucial to the execution of the VTCCCS.
5. **Other Nonfunctional Requirements**

5.1 **Performance Requirements**

Every action using the Tool should be no delays, and every operation involving the database will be performed in less than 2 seconds.

5.2 **Safety Requirements**

If there is a problem with internet connection, there may be potential data loss or errors when attempting to execute projects.

5.3 **Security Requirements**

New users and administrators are required to login with registered accounts. As of this time, The Tool uses object oriented methods to handle user data.

5.4 **Software Quality Attributes**

- **Functionality:** There are tools for users to use to edit, save, execute, and monitor project information.
- **Usability:** Navigation and project editing is clear and not complicated. QoS of web service information is also presented clearly for users.

5.5 **Business Rules**

At this time, there are no Business Rules set in place.
6. **Other Requirements**

At this time, there are no Other Requirements.
### Appendix A: Glossary

This Glossary is repeated from Section 1.4

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTCCCS</td>
<td>Visualization Tool for Composition of Cloud Computing Services</td>
</tr>
<tr>
<td>User</td>
<td>Someone who interacts with the client-side application</td>
</tr>
<tr>
<td>Admin/Administrator</td>
<td>System administrator who is given specific permission for managing and controlling the system</td>
</tr>
<tr>
<td>Node</td>
<td>A service generated by the user that will send or receive information from an external/internal web service</td>
</tr>
<tr>
<td>View Block</td>
<td>This is where nodes will be displayed depending on the View selected</td>
</tr>
<tr>
<td>View</td>
<td>Determines how nodes shall be displayed to the user</td>
</tr>
<tr>
<td>Red/Cyan</td>
<td>The state of node when toggled</td>
</tr>
<tr>
<td>Tool</td>
<td>Refers to VTCCCS</td>
</tr>
<tr>
<td>REF</td>
<td>Reference to something, etc. Eg.) Functional Requirement, FR</td>
</tr>
</tbody>
</table>
Appendix B: Analysis Models

MySQL Database Extended Entity Relationship Diagram
Appendix C: To Be Determined List
Third party services that will be supported by the tool.