Software Requirements Specification

Version 1.4

Power Usage Monitoring from UPS, PDU and hosts for a Data Centre Company

Visuallux: web based remote power consumption monitoring tool

Customer: Patrik Arlos
Software Development CEO: Dragos Ilie
Software Development Team: NetFix

TEAM MEMBERS

Student	Name	E-mail
1	ABIY BIRU CHEBUDIE	abch14@students.bth.se
2	ADDISHIWOT TADESSE	adtb14@students.bth.se
3	ELNAZ EKRAMIAN	elee10@students.bth.se
4	MEHRDAD MEHRABAN	memb10@students.bth.se
5	MOUNIKA REDDY CHANDIRI	moch15@students.bth.se
6	SUNEEL SEELAM	suse15@students.bth.se
7	TINASHE KUREHWASEKA	tiku14@students.bth.se
8	UDAYA BHASKAR GOLLAPALLI	udgo15@students.bth.se
9	VENKESH KANDARI	vekb15@students.bth.se
10	VIRINCHI BILLA	vibi15@students.bth.se

1. Preface

This is fifth release following initial, second, third and fourth release of the Software Requirements Specification. This report is written as a possible solution to customer's demand to monitor and control power usage in devices at a data centre company. This document targets the customer and the company involved in developing solutions to the customer's problems. The company comprises of Chief Executive Officer (CEO) and a team of developers called Netfix.

The structure of this document is organised that it starts with glossary and abbreviations which define the technical terms and abbreviations used in the document. Following is the section of system architecture which shows a high-level system design of system components interactions complemented by descriptions of the major components or modules of the system. Next is the requirements section which is divided into two subsections which are user requirements and system requirements. The final section is the references section showing a list of the references used in the preparation of this document.

Version 1.4-June 01, 2015
-Updating RESTful API service under module 3 and updating figure 4.

Version 1.3-May 20, 2015
-Adding RESTful API service under module 3

Version 1.2-May 14, 2015
-Adding RESTful API on high level architecture
-Adding login authentication and notification (email,sms,traps) requirements.

Version 1.1-May 05, 2015
-Adding RESTful API and updating user requirements and requirements table

Version 1.0-April 27, 2015
-initial release

2. Glossary and abbreviations

UPS - Uninterruptible Power Supply
An electrical apparatus that uses energy stored in batteries to provide emergency power to a load when the mains input power source fails.
PDU - Power Distribution Unit
A device for controlling electrical power in a data centre
CN - Computer Node
A computer that is connected to a computer network with a unique identity
SNMP - Simple Network Management Protocol
An internet standard protocol for managing devices network devices (routers, switches, servers, workstations, printers, modem racks and more) on IP networks.
HTTP - Hypertext Transfer Protocol
A set of rules for transferring files (text, graphic images, sound, video, and other multimedia files) on the World Wide Web
APC American Power Conversion
The leading manufacturer of UPS systems and surge suppressors
OS - Operating System
A program that manages other programs (applications) on a software running device
CEO - Chief Executive Officer
The most senior corporate member or administrator in charge of managing a for-profit organization or project.
UR – User Requirements

SR – System Requirements
FUR – Functional User Requirements
FSR – Functional System Requirements
NFUR – Non Functional User Requirements
NFSR – Non Functional System Requirements
T?M? – Test ? Module ?: (? is replaced by a counting numbers)
Test identification string

3. System architecture

This section illustrates a complete overview of the system architecture high-level design. Within this section are subsections describing and illustrating the major components or modules of the system

3.1. General overview (overall architecture)

The software product's main task is to enable the user to monitor the power consumption of remote network devices present in the data centre network, monitor the UPS battery level and to allow the user to export and import data through RESTful API. The remote devices to be monitored include but are not limited to UPSs, PDUs and hosts. Figure 1 shows an overview of the networks structure where the software product is installed and run on the server. The software tool is composed of three main modules which are the backend, database and frontend which are described below.

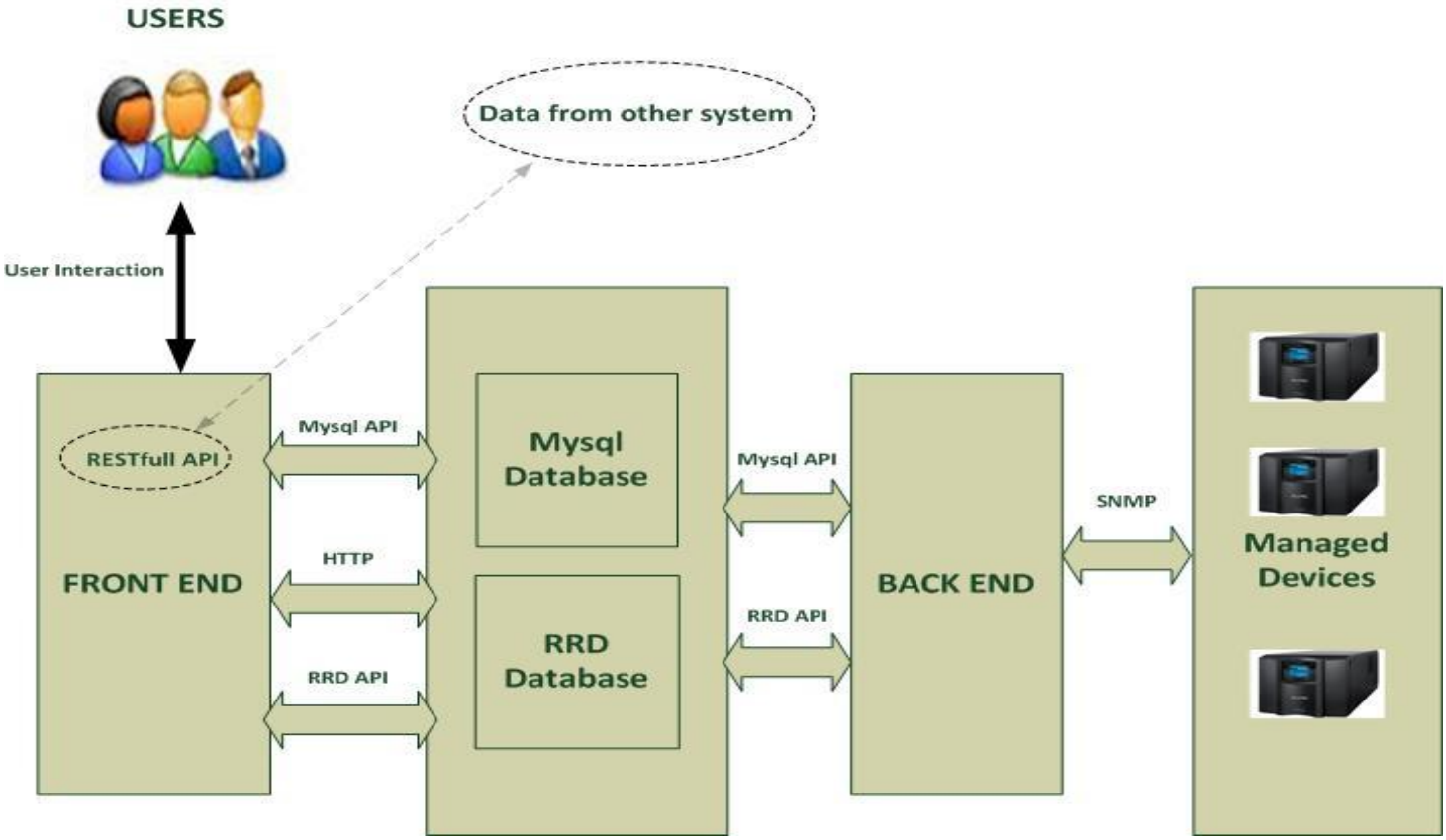

Figure 1-system architecture

3.2. Module 1: Data retrieval tool (backend)

The data retrieval tool is the one which directly communicate with the devices to be monitored. It gets the device identities entered by the user in the database and, it uses SNMP to poll the devices for the information concerning power consumption and UPS battery level, and store the received responses in the database. The backed has notification services about of the battery level of UPSs (critical, danger and normal). Once the software is installed on the server and running, the data retrieval process runs continuously to get information of all devices in the network and update the database every one minute. Figure 2 illustrates the description of the backend.

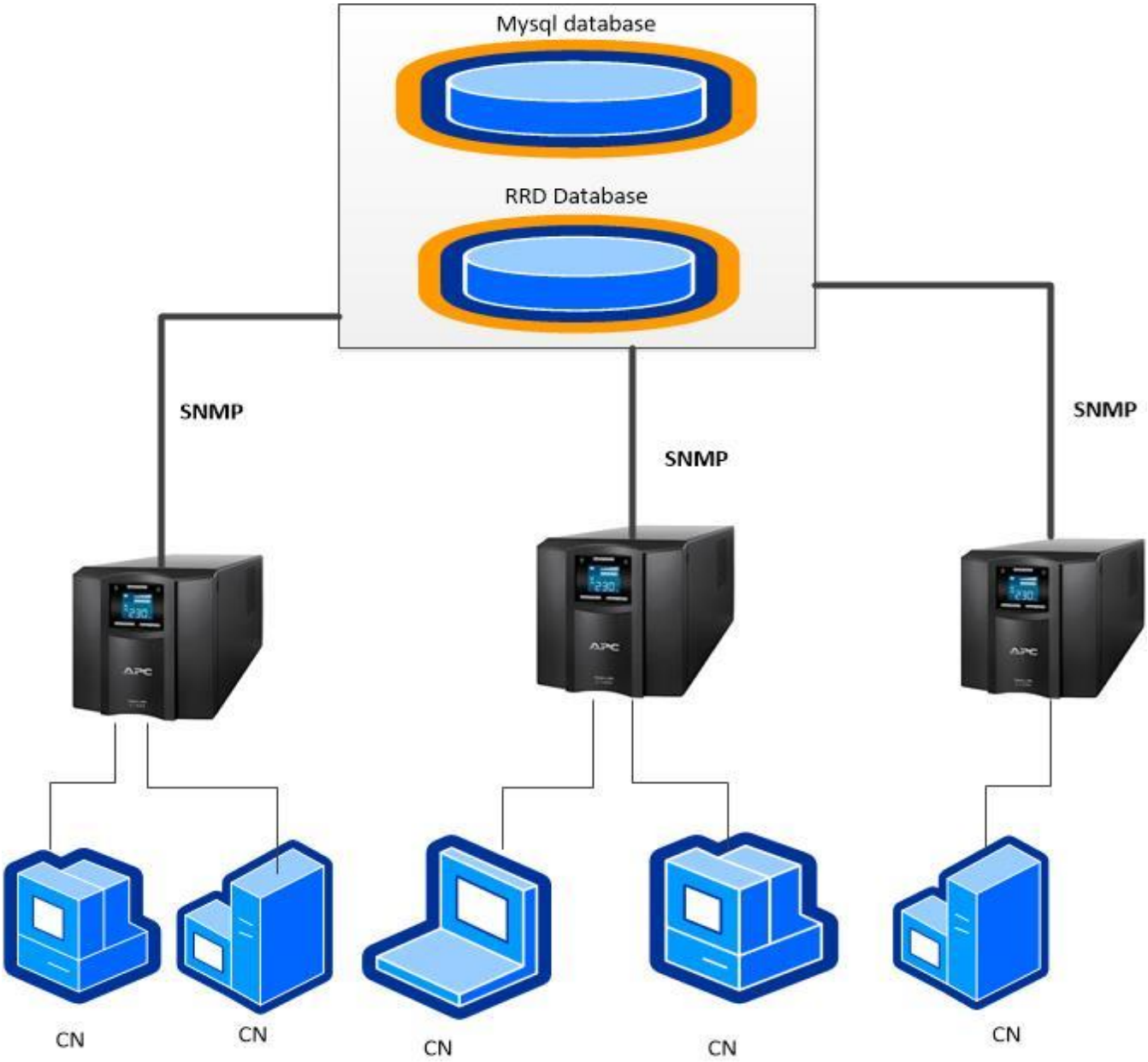

Figure 2-backend

3.3. Module 2: Data storage component (database)

The database is responsible for the storage of different types of information. The MySQL database contains device details which the backend uses to connect to the devices and poll for power consumption information. The responses from the devices is stored in the RRD database and accessed from the frontend to give graphs of the device statistics. The database is located in the Linux server where the backend (data retrieval tool) runs as to be easily accessed the backend to get the devices to poll from MySQL database and stored the response in RRD database such that it is accessible in different ways depending how the user wants to view the device statistics. The MySQL and Round Robin Database are responsible for the interaction with both the backend and frontend. Figure 3 illustrates the description of database.

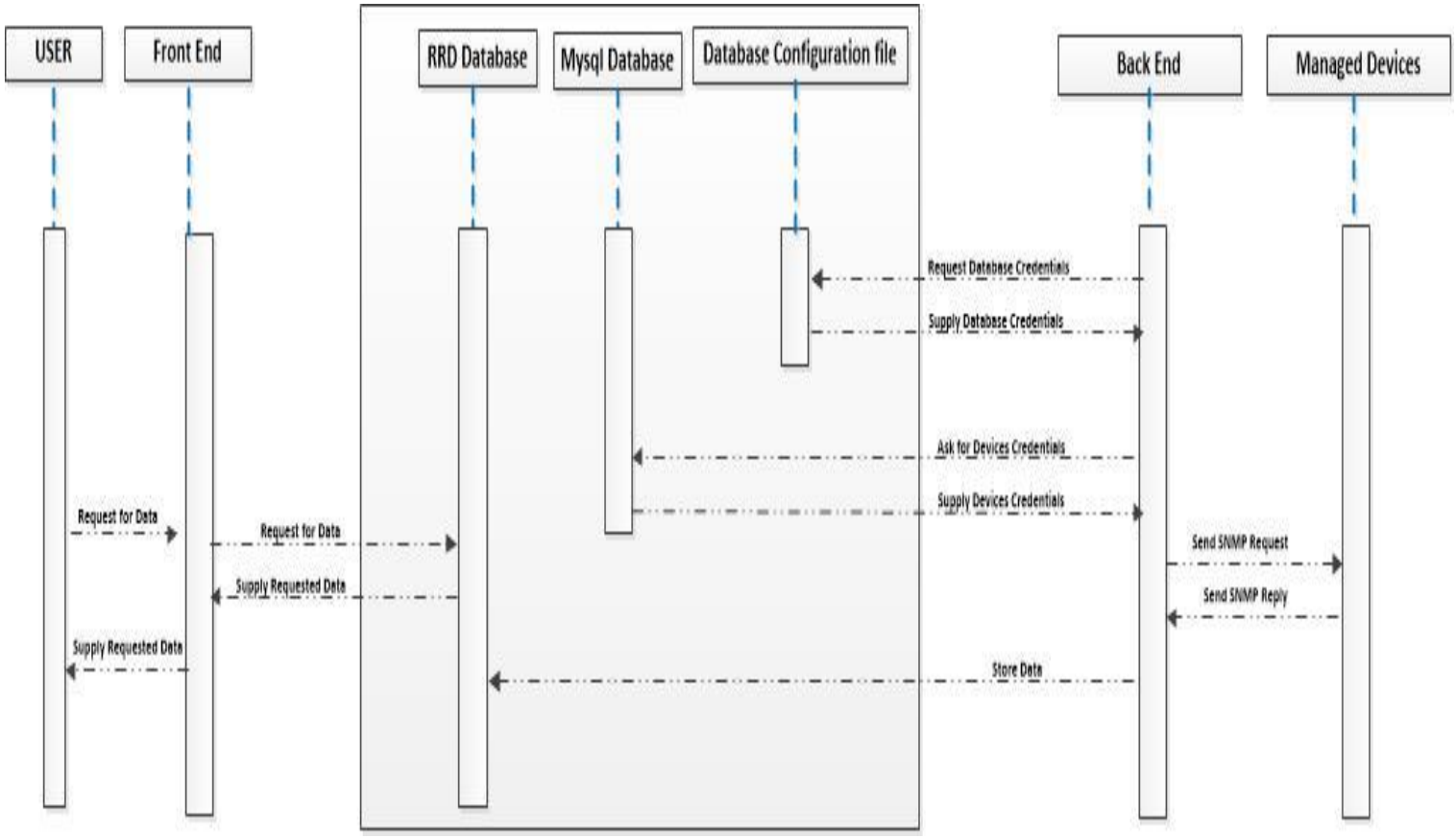

Figure 3 - Database

3.4. Module 3: User interaction interface (frontend)

The frontend is a web graphical user interface which interacts with the user. From front end, the user can add, edit, select and/or remove a device to be monitored and, generate power consumption graphs at different time scales for a selected device using RRDtool. This module offers the RESTful API service which allows the user to import and export data from and to another application and/or system. The data exchange will be done through http requests using crafted urls. The data to be exchanged through the software tool is encoded in JSON format hence the client to request the data must have a JSON format decoder. The data format is represented as key and value. For the RESTful service of the software product, each data value has a time stamp as its key in the JSON exchanged data. The power monitoring data sets will be in the format of "<time stamp> <data>" and will be provided per PDU and per UPS. Each PDU/UPS is identified by IP, PORT, and COMMUNITY hence for exporting the power

data of a single UPS/PDU under monitoring to another system (the REST client), the user of another system must request the data through a web browser by using a url in the format http://<serversIP>/rest.php/?req=power.<deviceIP>.<devicePORT>.<deviceCOMMUNITY>. The exported data supplied by the software tool is within the time range of within the past 12 hours. For importing data from another system the software tool requires the user to enter a url which is specified by the server (REST server) providing data. The dada format accepted by the software tool (REST client) must be in JSON format as the tool uses a JSON decoder. The data must be presented in the format of "<time stamp> <data>" and the software tool will present the data in graph. The frontend is installed on the Linux server and is accessed by a web browser. An illustration of the frontend shown by figure 4.

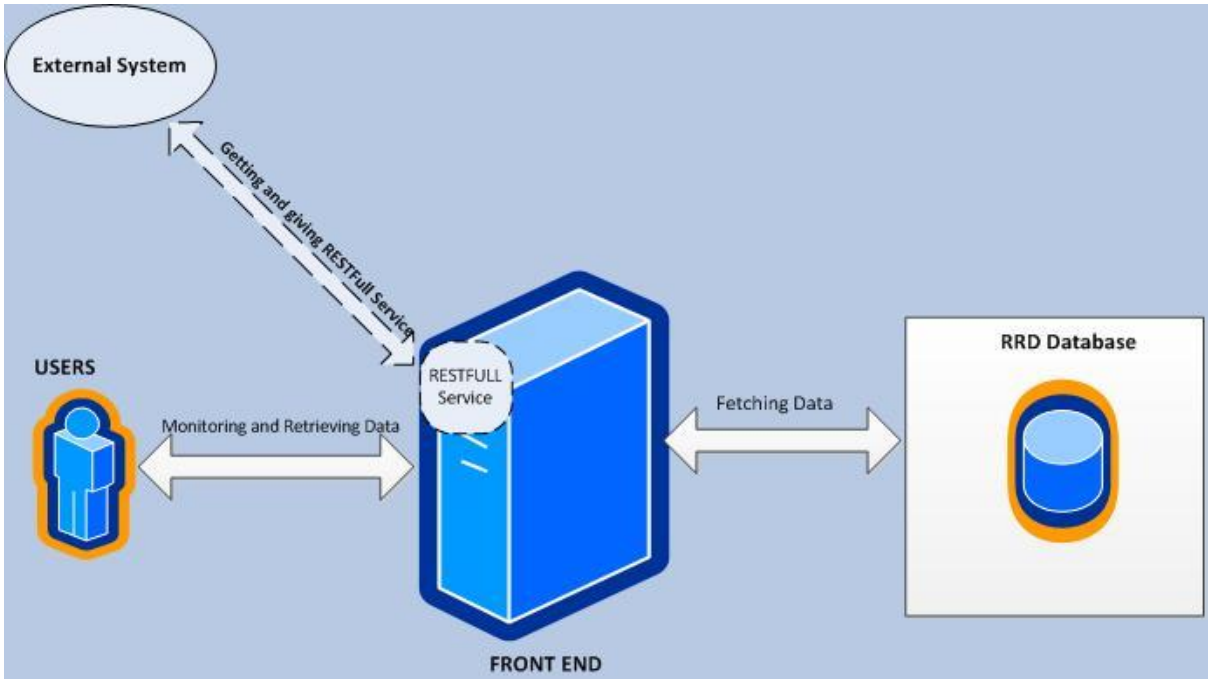

Figure 4 – Frontend

4. Requirements

This section is divided into two parts which are the user requirements and the system requirements. The user requirements are what the customer expect from the software tool for validation of the *acceptance test plan document*. System requirements are technical requirements that complement the user requirements and provide enough information for design and implementation.

4.1 User requirements

This section describes the services that the user get from the software product tool. The user requirements are classified as functional user requirements (**FUR**) and non-functional user requirements (**NFUR**).

4.1.1 Functional User Requirements

FUR 1: Monitoring power usage at UPS level

This is one of the primary functionality the software tool required by the customer. The software must allow the user to monitor the power usage of UPSs in the network of a Data Centre company.

FUR 2: Monitoring power usage at PDU level

This is one of the primary functionality the software tool required by the customer. The software must allow the user to monitor the power usage of PDUs in the network of a Data Centre company.

FUR 3: UPS battery monitoring

The software tool must allow the user to monitor battery remaining time and/or battery remaining capacity of UPSs in the network of a Data Centre company.

FUR 4: Adding devices

The software tool must provide the user with a functionality to add devices to be monitored without any problem while it is running.

FUR 5: Removing devices

The software tool must provide the user with a functionality to remove devices to be monitored without any problem while it is running.

FUR 6: Adding and editing users

The software tool must provide the user (admin) with a functionality to add remove edit other users of the software (admin, user and guest).

FUR 7: Scalability

The software product must handle a large number as the customer is an expanding data centre company with future plans of increasing number of devices to be monitored.

FUR 8: Vendor independent

The software tool must retrieve power consumption data of different devices regardless the manufacturer (vendor) as customer (Data Centre Company) purchases its equipment from different vendors.

FUR 9: Display correlation of data sets on a time series graph

From a web browser interface accessible frontend, the software tool allows the user to select a device and display a graph to correlate the data sets associated with that device. For example if the device is a PDU, its status, power usage, battery state, battery remaining time and battery temperature are displayed. The graph display shows the monitored device metrics with respect to time which is

presented in different scales of day, week, month and year as well as text details showing the current values of the metrics. The graphs will be presented in all mentioned time scales and not selectable.

FUR 10: Display power maps of devices

As required by the customer, the Visuallux network device power monitoring software tool provides a power map display for all the devices which shows to which other devices is the displayed device connected to. This feature is provided from the frontend and it is useful to quickly know which other devices will be affected if there is a power failure of either a UPS or PDU.

FUR 11: RESTful API

The software tool must have a RESTful APIs which allows it to can communicate with other applications software and/or systems to export and import (exchange data). The exchanged data can be will be in JSON format hence the software's client must have a JSON decoder. And will be passed through the use of http requests using urls. The imported data from the other systems will be processed and presented in a graphical format on the web GUI.

FUR 12: User interaction web interface platform

As a customer's requirement, the software tool (Visuallux) must provide web interface accessible frontend that allows the user to add, edit or remove the devices to be monitored as well as requesting the power usage statistics of specific devices or group of devices to be displayed as text or images (tables or graphs). See FUR 9 for display and FUR 10 for groups.

FUR 13: Login authentication

The software must allow the user to enter login credentials to access the frontend functions for viewing, adding, removing and editing the devices under monitoring and the fetched data.

FUR 14: Notifications (email,sms,traps)

The software tool must allow the user to receive on another device, threshold notifications (normal, warning, critical) of battery and/or power status when there is power failure in PDU. The notifications are either email, sms or traps and must send before the UPS battery fails in order to allow the user to take protective measure on the saved devices before they run out of power.

4.1.2 Non Functional User Requirements

NFUR 1: User friendly

The software product provides a user friendly web GUI to facilitate easy power monitoring of devices. The software tool is handed to the customer with all the necessary user manual document containing information about installation and usage.

NFUR 2: Upkeep

The software product comes with maintenance support and documents citing possible future faulty, errors and risks and the immediate action to take in reaction to the events to minimise damage of the network system.

4.2 System requirements

This section describes technical requirements that complement the services that the user get from the software product tool. The system requirements are classified as functional system requirements (**FSR**) and non-functional system requirements (**NFSR**).

4.2.1 Functional System Requirements

FSR 1: Operating System

The required operating system of the server on which the software runs is Ubuntu 14.04 LTS. The desired RAM for the operating system is 2GB.

FSR 2: Web Server

A web server (open source apache webserver) is required to be installed on the server which runs the software tool.

FSR 3: Database

A MySQL database and a time series open source RRD database are required to be installed on the server which runs the monitoring software tool.

FSR 4: Programming language modules and libraries

Several programing language modules and libraries are required to be installed on the server on which the monitoring software too runs. These include snmp, rrdtool database, mysql database modules and libraries for perl and php.

FSR 5: RESTful API and other APIs

An API to support RESTful function of importing and exporting data must be installed on the server.

4.2.2 Non Functional System Requirements

NFSR 1: Performance

A server with acceptable performance is required for the software tool to hold the capabilities to perform the tasks required to meet the acceptance test plan.

NFSR 2: Security

A username and password security system is required on the server for the authorised users to get access databases with the details of devices under monitoring.

4.3 Table of requirements

Requirement	Creation Date	Change Date	Module	Type	Test
FUR1	27-04-2015	05-05-2015	1	Functional	M1T1
FUR2	27-04-2015	05-05-2015	1	Functional	M1T2
FUR3	27-04-2015	14-05-2015	1	Functional	M1T3
FUR4	27-04-2015	05-05-2015	3	Functional	M3T1
FUR5	27-04-2015	05-05-2015	3	Functional	M3T2
FUR6	27-04-2015	05-05-2015	3	Functional	M3T3
FUR7	05-05-2015	-	1	Functional	M1T4
FUR8	05-05-2015	-	1	Functional	M1T5
FUR9	05-05-2015	-	3	Functional	M3T4
FUR10	05-05-2015	-	3	Functional	M3T5
FUR11	05-05-2015	-	3	Functional	M3T6
FUR12	05-05-2015	-	3	Functional	M3T7
FUR13	14-05-2015	-	3	Functional	M3T8
FUR14	14-05-2015	-	1	Functional	M1T6
NFUR1	27-04-2015	-	3	Non Functional	-
NFUR2	27-04-2015	05-05-2015	1,2,3	Non Functional	-
FSR1	27-04-2015	-	1,2,3	Functional	M1T7, M2T4, M3T9
FSR2	27-04-2015	-	3	Functional	M3T10
FSR3	27-04-2015	-	2	Functional	M2T5
FSR4	27-04-2015	14-05-2015	1,2,3	Functional	M1T8, M2T6, M3T11
FSR5	14-05-2015	-	3	Functional	M3T12
NFSR1	27-04-2015	-	1,2,3	Non Functional	-
NFSR2	27-04-2015	-	2	Non Functional	-

Table 1 - requirements

5. References

1: Software Development for Telecommunication lectures and itslearning: Dr Dragos Illie, BTH 2015
2: Applied Network Management Seminars and Labs: Dr Patrick Arlos, BTH 2015
3: Google images