Project Specification - System status portal

Table of Contents

[1. Document description 3](#_Toc209618340)

[1.1. Purpose 3](#_Toc209618341)

[1.2. Objective 3](#_Toc209618342)

[1.3. Project Responsible 3](#_Toc209618343)

[2. Requirements 4](#_Toc209618344)

[2.1. Scope 4](#_Toc209618347)

[2.2. Limitations 5](#_Toc209618348)

[2.3. Main Tasks 5](#_Toc209618349)

[3. Project Organisation 6](#_Toc209618350)

[3.1. Internal Resources 6](#_Toc209618351)

[3.2. External Resources 6](#_Toc209618352)

[4. Project Status and Reports 7](#_Toc209618353)

[4.1. Report Frequency 7](#_Toc209618354)

[4.2. Distribution List 7](#_Toc209618355)

[5. Abbreviations 8](#_Toc209618356)

[6. Appendix A 9](#_Toc209618357)

Figures

[Figure 1: Example world map view. 4](#_Toc209605888)

1. Document description
   1. Purpose

Today we make monthly reports on every installed system with a production volume of over 100 samples and every quarter for all other systems in active use. Report generation has been automated from data manually downloaded and stored in a local database. Since this is done on a monthly or quarterly basis, we usually don’t see the issues with the system until weeks after the fact and then it takes lots of time to analyze logs and it is difficult to get good answers from operators on what happened. With continues live information on the state of all the systems at a quick glance, we will have critical information faster and be able to find problems quicker and with less investigation time.

* 1. Objective

The outcome of the project is to generate a web portal that presents data from different sources, and with different views depending on receiver and type of data.

The project will be divided into different stages depending on functionality.

1. Web server and world map presentation
   * The initial stage will be proof of concept and run locally on a laptop.
2. Create an API to access data on remote system
   * Initially on lab system at SinterCast Technical Center
   * Only Windows system information
3. Set up a database to store data acquired from system.
4. Expand with more system information (e.g. PCS alarms, Tracking alarms, Process events)
   * This part must be specified depending on the outcome of the first 3 parts.
   1. Project Responsible

Roger Andersson

1. Requirements
3. 1. Scope

* Identify an open-source web server that supports dynamic web pages. The server must be self-hosted and Windows and Linux.
* Create a simplified world map with customer sites marked out. Each marker should have color coding to show status of the system according to specified parameters.

Figure 1: Example world map view.

* Color coding

|  |  |  |
| --- | --- | --- |
| **Color** | **Status** | **Comment** |
| Grey | Not connected to SOSON | No data will be presented |
| Black | System is not accessible | Link to system is down |
| Green | System is active with no alarm | All states are within limits |
| Yellow | System is active with warnings |  |
| Red | System is active with errors |  |

* Data retrieved from remote system must have good/bad or limits configurable in the web application (receiving end) to determine state. The criteria’s for the states does not have to be configurable for each system.
* When clicking on a site following data should be presented in a pop-up dialog on the main page, information is shown in a table format.
* Following data should be presented in the initial part, but the view must be expandable to add more information.
  + API version
  + Windows version
  + Uptime
  + Computer name
  + Actual time
  + Time zone
  + Daylight saving
  + Disks present (Mount point, Name and Type)
  + Disk utilization
  + Memory usage
* API to present the data from remote system side. See Appendix A.
* Set up a database to store information fetched from systems. In case link is lost then the last data received is accessible. Data must be stored in the database for 2 months.
* Expanding with more views and process software data will be added on at a later stage, however it should be considered during development of the first stage.
* GIT must be used for revision handling of code.

Recommended applications and methods to use

* Programming language
  + C#
  + Python
* Database
  + PostgreSQL
  + MSSQL
* Web page generation and data exchange
  + HTML
  + JSON
* Interface to access data
  + REST API
  1. Limitations
* Retrieval of data will only be performed on systems connected to SOSON.
* Applications and code libraries used in the project should only be open source with a MIT like license.
  1. Main Tasks
* Develop a web application to present data retrieved from remote systems run on a local web server.
* Install and configure a database to store data.
* Develop an API for the requested data.
* Documentation. Function description of program code, installation and setup.

1. Project Organisation
   1. Internal Resources

Roger Andersson

Philip Miranda

Ulf Norman

Samuel Jansson

* 1. External Resources

N/A

1. Project Status and Reports
   1. Report Frequency

Status report before the 30th each month.

* 1. Distribution List

1. Abbreviations

API Application Programming Interface

REST Representational State Transfer

PCS Process control software

SOSON System on SinterCast network

HTML HyperText Markup Language

JSON JavaScript Object Notation

SQL Structured Query Language

MSSQL Microsoft SQL

1. Appendix A

