SySTemantics Design Document

By

Dixita Sharegar

Bhargav Uppalapati

Siva Chintapalli

**Sponsor**

Anthony Giorgio

Poughkeepsie, NY

February 2016

# **Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Author | Date | Status & Description |
| 0.1 | All | 02/25/16 | First Draft |
| 0.2 | Dixita | 03/06/16 | Document history section added |
| 1.0 | All | 03/10/16 | Final Draft |

# **Audience**

This document is intended for designers, developers & engineers who want to modify or extend the existing implementation of the metrics collector application. It is also intended for customers who want a detailed description of the system.

# **Objective**

A Metrics collector is a console based application which runs on the Linux-based operating system. The application incorporates CPU stats, memory stats, Network stats. It helps users of the system to manage the CPU throughput, and view network and memory stats allocated to the applications. Keeping track of all the system matrices without a proper application is hard. It is also very difficult to remember the commands to view all the system statistics. The statistics are logged into a database frequently which can be viewed by the user to keep track of system performance.

# **Approach**

The main goal of the project is to give the accurate and reliable information of the System Statistics.

· Goal 1: Implementing the database.

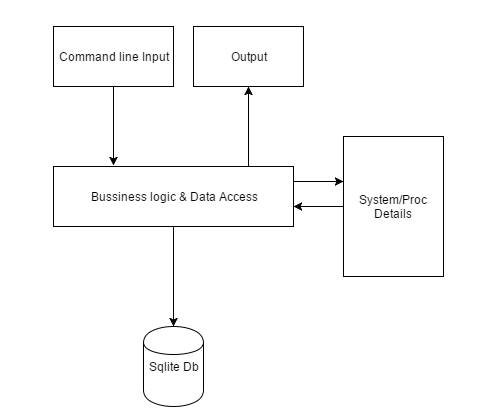
· Goal 2: Designing the User Interface.

· Goal 3: Implementation (Connection to the database and logical Operations)

· Goal 4: Displaying the records/information in a report format.

· Goal 5: Testing and maintenance of the project.

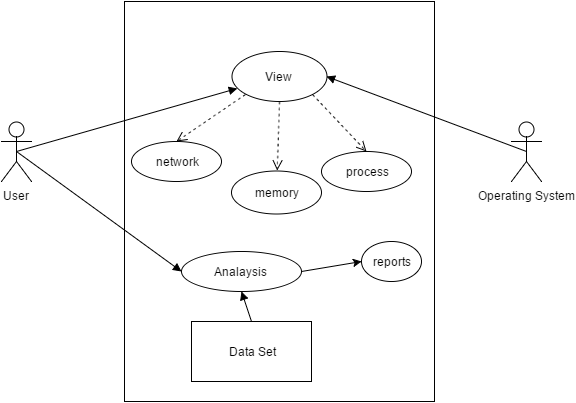
**ARCHITECTURE**



# **External Design**

## Command Line Interface (CLI)

# **Use Case Diagram**

 Application

1. User requests the application to view the running tasks/process in the system.
2. The operating system will control the application to provide the required information.
3. The user can view network/memory/process statistics.
4. The user can view and print the reports based on the time constraint.

# **Internal Design**

## Development Standards

* The project involves usage of agile methodology.
* Feedback of end-user is taken at every phase and developed accordingly.
* Java, C and SQL languages are used in a development of SySTematics.
* ncurses for interfaces for the editor.
* For graphs Gnu plot, Graphite, Darkstat.
* SySTematics will be open-source software which will be available to the public.

## Hardware Resources

* Physical machines : Desktop
* Operating systems : Linux
* Application Type :Console Based Application

## Development environment:

* Compilers : GCC,JVM
* IDE : NetBeans
* Source code repository : GitHub
* Build process : Maven
* Database : SQLite
* Instance : Single

# **Software flow**

The installer will install all software packages needed for our application.

Metrics will be collected using different Linux commands.

Memory, CPU and network metrics will be aggregated and displayed in common units.

The collected metrics will then be stored in the SQLite database.

Data is fetched from the database based on what the user is viewing. End User will not be aware of the processes above this block

All related metrics will be displayed to the user based on his screen selection.

The application will terminate and the data collection and storage in database will stop.

## Error Handling & Recovery

The application will log every step or function call, so that it is easy for the developer to duplicate the issue and solve it. If one part of the application is not working properly, it will fail gracefully and an error message will be displayed on screen.

## Testability

Test Environment will be same as the customer’s environment i.e. Linux OS, Standalone machine.  
Test Scenarios include testing with heavy CPU processes, memory consuming processes, I/O processes etc.  
Regression Testing will be performed after every module is completed to ensure all old functionalities are working fine. Test cases will be written to record all results.

Function testing and system testing will be performed after completion of every module.

## Packaging

An installer will be used to install and uninstall the application.

**Data**:

This application integrates SQLite database:

* Which dynamically stores CPU stats, memory stats, Network stats.
* The need for using SQLite database is its support for many programming languages including java.
* The data update frequency is 5 seconds.

**Scope**:

Outcome of the project:

* The objective of this application is to provide the user a clear picture of system performance.
* Any user can run this system and access the data from the database.
* History as old as one week is maintained in the database and can be viewed/retrieved at any time.
* The database maintains the current and past information.
* Network stats, CPU stats, memory stats are represented by visual indication.
* Users can print the logs of the system statistics but cannot change the data.

# **Security**

There is no requirement for the software to run as root, but can be run as root authority if needed.

* Any user with the installation files can install and use the software. If needed, admin rights can be given.

# **Accessibility**

* The basic requirement, amongst other things, that there is sufficient contrast between text and background color.
* Best practice is to avoid the usage of red/green colors. As 1 in 12 people have red/green /yellow color disability.

# **Globalization**

* The UI buttons, display messages, console outputs are in English.
* SysTemantics supports only English language.
* Documentation is in English.

# **Supporting Material**

## Glossary

## References

**PROJECT BUDGET:**

* No Project Expenses

# **Risks and Dependencies**

## Risks:

* Manual Testing is performed throughout the project (Automation testing is not performed).
* It is designed as a desktop application.

## Constraints:

* Lack of mobile and web application design implementation.
* Time.

# **Assumptions:**

* All the team members are dedicating their time on completing their tasks.
* The design document may be modified based on the project requirements.