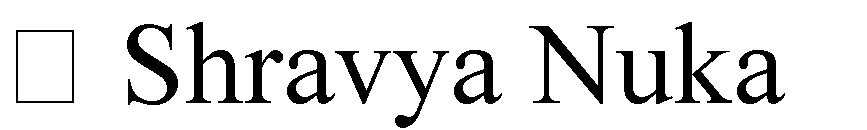
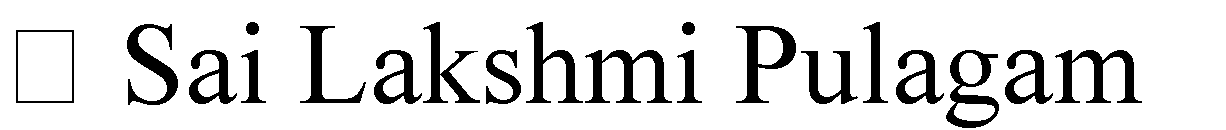
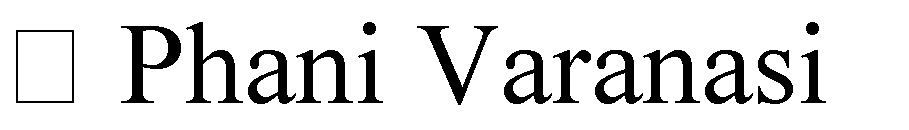
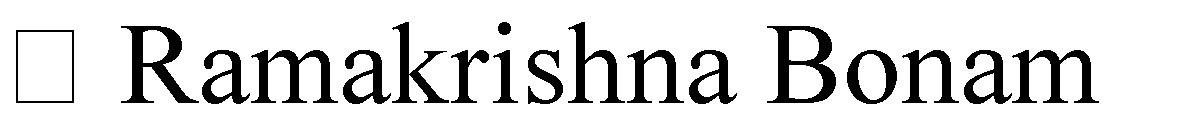
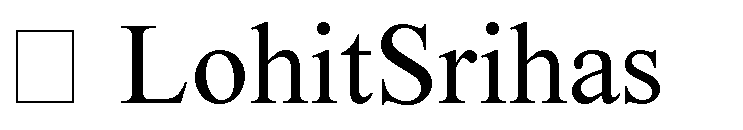
**OVERLAY NETWORK MONITORING**

Project Specifications

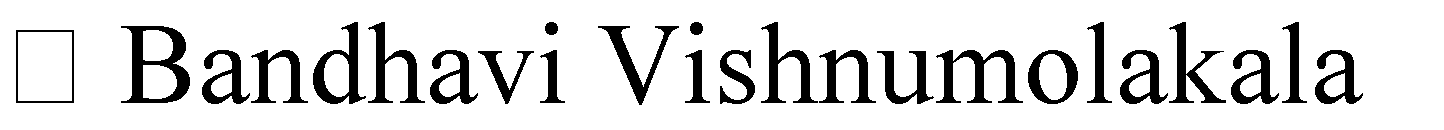
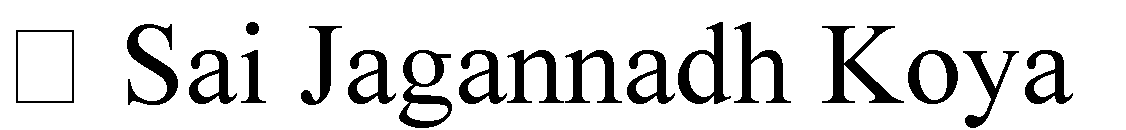
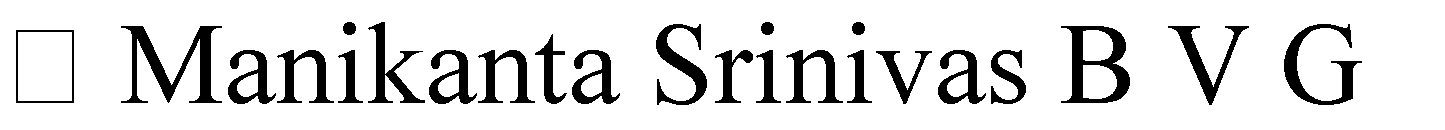
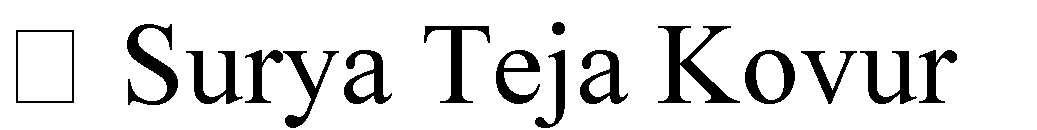
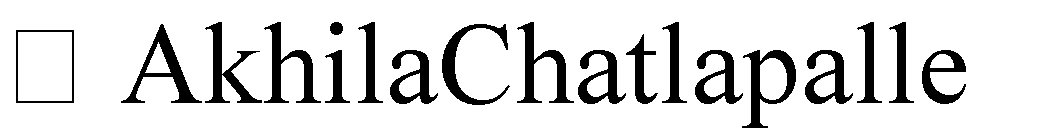
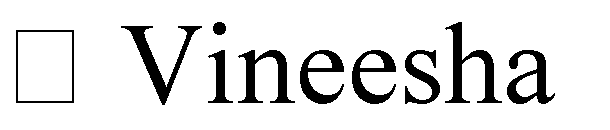
Version 1.3

**Team Name:** Smart Developers

**Team Members:**



Sana



1. **Preface:**

The main concern of the project is to provide the customer a simple and unified way of maintaining and updating its applications which interact with the monitoring system through a common RESTful API. This is the initial version of the document.

Release v1.3 on 2017-06-16:

* Made few layout modifications

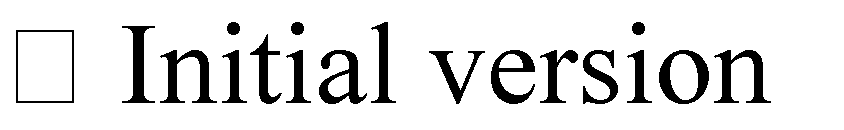
Release v1.2 on 2016-12-18:

* Made changes in the section glossary and abbreviations.
* Made changes in work management.

Release v1.1 on 2016-05-01:

* + Made changes in the proposed solution by modifying design.
  + Added relevant limitation.
  + Updated Time plan by using gantt chart.
  + Made changes in Configuration Management.
  + Made changes to the previous explained method of progress tracking.
  + Quality control updated.
  + Made changes to Risk Management.
  + Few changes done in System Release plan.

C. *Release version 1.0 on 2016-04-17:*



In the remainder of the document, Section II describes briefly about the basic abbreviations used in this document. Section III describes the background, specifying requirement of the customer and process implemented by the developers. Section IV describes the functionality of the proposed tool. Section V specifies the scope of the project. Section VI specifies the limitations of the project. Section VII dictates the time plan of the project, and tollgates are mentioned. Section VIII shows the project organization among the members of development team, work is broken down and each division assigned to different members of the team. Section IX describes the Configuration Management which involves managing versions, system building and release management. Section X shows Progress Tracking by assigning checkpoints. Section XI gives Quality control measures and ensures effective execution of tool. Section XII defines Risk management. Section XIII gives an estimate of System Release Plan, which involves testing, packaging and documentation plans. Lastly, Reference section cites references used for documentation and tool development.

1. **Glossary and Abbreviations:**

* **API**: Application Programming Interface
  + An API is a set of routines, protocols, and tools for building software applications.
* **InfluxDB**: Influx Database
  + InfluxDB is an open source time series database. InfluxDB has no external dependencies and provides an SQL-like language with built in time-centric functions for querying a data structure composed of measurements, series, and points.
* **Grafana**:
  + Grafana is most commonly used for visualizing time series data for Internet infrastructure and application analytics but many use it in other domains including industrial sensors, home automation, weather, and process control. Grafana features pluggable panels and data sources allowing easy extensibility and a variety of panels, including fully featured graph panels with rich visualization options. There is built in support for many of the most popular time series data sources.
* **SSL**: Secure Sockets Layer
  + SSL is a standard security technology for establishing an encrypted link between a web server and a browser.
* **RTT**: Round Trip Time
* **HTTPS**: Hyper Text Transport Protocol Secure
* **RESTful**: Representative State Transfer

1. **Background:**

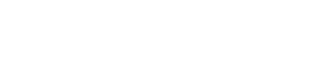
The company Smart Net (the customer) develops distributed applications that require QoS measurements to adapt themselves to the network conditions. Smart Net has determined that the situation makes it difficult for each of their application to perform network measurements individually. To overcome this

situation we would like to design an overlay network monitoring system through common RESTful API that will have a unified solution.

1. **Proposed Solution:**

The customer will be provided with a full mesh overlay network monitoring system to interact with the applications developed by SmartNet. It consists of admin server which establishes a HTTPS connection with three or more peering nodes. We develop an interaction user-to-admin server & node to admin server

that is based on a RESTful API. We collect the historical data in the form of graphs for the given time slots.



RESTful API



1. **Scope:**

Overlay networks will provide a viability of network utilization. Overlay approach decreases the number of delayed packets in wide area reliable communication leading to improve performance in time sensitive applications such as voice over IP

1. **Limitations:**
   1. As HTTPS is used for encryption and decryption it results in time lag.
   2. Too many requests from the clients lead to congestion.
2. **Time Plan:** Mile stones

April15-17

Project specification

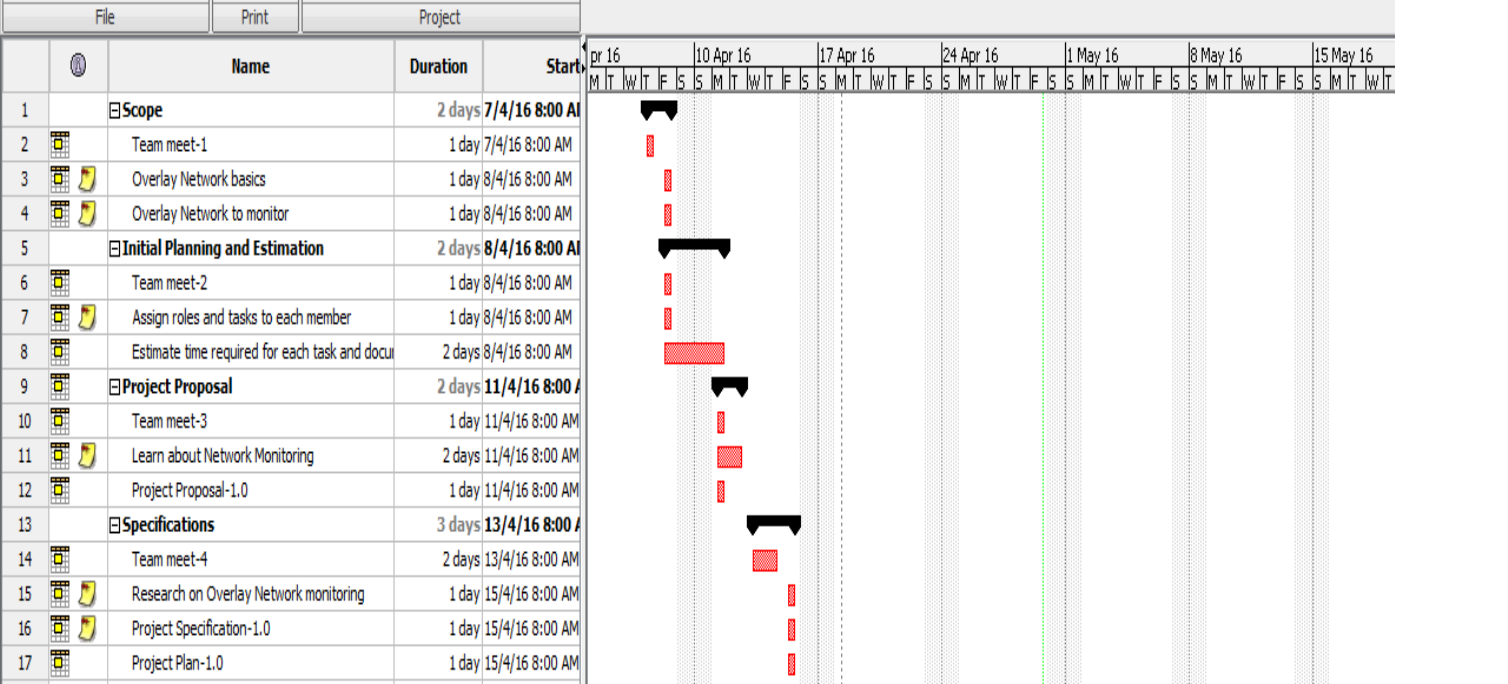
May15-22

Product release

April 17-24

SRS

April24- May1



Design

May1-8

Accept plan

May8-15

Release candidate

Toll Gates:

1. 17-04-2016: Project proposal

2. 17-04-2016: Project Specification

3. 24-04-2016: Software Requirements Specifications

4. 01-05-2016: Design document

5. 08-05-2016: Acceptance test plan

6. 15-05-2016: Project Documentation

7. 22-05-2016: Product Release

1. **Project organization:**

|  |  |
| --- | --- |
| Programming  Frontend:  Database:  Backend: | Akhila Chatlapalle  Ramakrishna Bonam, Vineesha Sana  SaiLakshmi Pulagam  Shravya Nuka  Manikanta Srinivas  Phani Varanasi  Bandhavi Vishnumolakala,  Lohit Sriha Korlepara,  Manikanta Srinivas |
| Testing: | Manikanta Srinivas, Lohit Srihas Korlepara |
| Management: | Lohit Srihas Korlepara |
| Documentation: | Sai Jagannadh Koya,  Sai Surya Teja Kovur |

Table 1: work management

1. **Configuration Management:**

9.1 Version Management:

This method of adding tags will avoid confusion while committing branches. The client servers will clone the whole repository and if any server dies the client repositories will copy back the data to the server.

9.2 System Building:

System building is done using GNU make tool which controls the generation of executables and other non-source files of a program from the program's source files. While writing a program we will create a make file which enables the GNU to build and install program. Also the make file specifies the shell commands to compute it. These shell commands can run a compiler to produce an object file or the linker to produce an executable file.

9.3 Release Management:

The early build version of software is released to customer. The customer uses the product and reports the problems. The bugs are fixed before the next release and depending on the types of problems and bug fixes the release can be classified to major release and minor release. Each release is documented to ensure that it can be re-created exactly in the future. The source code copies must be maintained along with the versions of the libraries, compilers and tools used for development for future release purposes.

1. **Progress Tracking:**

We will maintain a schedule in ProjectLibre and will arrange meetings likewise. We will discuss the progress of each member individually and we will continuously monitor whether our team is on time or lagging behind.

1. **Quality Control:**

In order to maintain the expected quality of the customer, the testing wing of the group plays a vital role. They will crosscheck the source code and given requirements of the project. They will go through the customer needs and verify it with our released product. The released product can be finally validated by testing and taking feedback from the customer and by making the relevant changes.

This involves:

1. Unit testing: individual program units or object classes are tested
2. Component testing: focus on testing component interfaces
3. System testing: focus on testing component interactions
4. **Risk Management:**

Risk management is the ability to predict the problems beforehand and taking necessary strategies to minimize them.

Risks can be of three types:

1. General risks
2. Risks due to hardware
3. Risks due to software

General risks:

|  |  |
| --- | --- |
| Risk | Strategy |
| Lagging behind schedule | Find out our weak areas where we are lagging and put more effort to  Enhance them. |
| Team members are sick | We will make necessary arrangements as to fill the gap and report the same  to the in charge. |
| Buying the necessary components | Whole team will contribute in taking decisions. |

Risks due to hardware:

|  |  |
| --- | --- |
| Risk | Strategy |
| Components may fail | Check each and every node and it’s  Connections. |
| Server may fail | Try to re-establish connections or replace in case of severe damage. |

Risks due to Software:

|  |  |
| --- | --- |
| Risk | Strategy |
| Change requirements | Asses the quality of product and remodel if necessary |
| Merging code | We will follow a version management system and merge the code from our team members thereby avoiding confusion. |

1. **System Release Plan:**

13.1 Testing plan:

At the end of product, Tests are made to detect bugs and ensure efficiency of the product. They are performed to check easy accessibility of the tool to the customer. The details are:

Frontend: 29/04/2017

Back end: 5/05/2017

If any bugs are detected,

Retesting: 8/05/2017

13.2 Packaging plan:

A compressed tar.gz archive is provided to the user consisting of the alpha code, library files, tools and related documentation. The details of the release plan are as follows:

Release candidate: 5/06/2017

Final release: 14/06/2017

13.3 Documentation Plan:

13.3.1 Installation Documentation:

The Installation document is released in a PDF format. It covers the installation procedures of different software components and also the configuration settings for different components.

13.3.2 User Documentation:

The user document is released in a PDF format. It covers the

* scope of the tool
* functionality of the tool
* linking different modules in the tool
* procedure for supplying inputs to the tool
* Generation of output and various scripts

**14. References:**

1.[http://www.bth.se/com/dil.nsf/attachments/ilie2010unicast\_qos\_routing\_in\_over](http://www.bth.se/com/dil.nsf/attachments/ilie2010unicast_qos_routing_in_overlay_networks_pdf/$file/ilie2010unicast_qos_routing_in_overlay_networks.pdf) [lay\_networks\_pdf/$file/ilie2010unicast\_qos\_routing\_in\_overlay\_networks.pdf](http://www.bth.se/com/dil.nsf/attachments/ilie2010unicast_qos_routing_in_overlay_networks_pdf/$file/ilie2010unicast_qos_routing_in_overlay_networks.pdf)

2. [http://www.download4referencebook.com/journals/overlaynetworks-](http://www.download4referencebook.com/journals/overlaynetworks-monitoring) monitoring