BLEKINGE INSTITUTE OF TECHNOLOGY

DESIGN DOCUMENT

TCP Evaluation in Semi-Live Streams

"Team Enigma"

Anirudh Kodaru

Hemanth Kumar Ravuri

Nandini Chowdary Godavarthi

Naren Naga Pavan Prithvi Tanneedi

Reventh Thiruvallur Vangeepuram

Sasidhar Podapati

Sathvik Katam

Srinand Kona

SriKavya Chavali

Vaibhav Bajaj

Venkata Sathya Sita J S Ravu

Version 1.4
Publication Date: 2015/08/24

1. PREFACE:

This is the Design Document of the project titled, “TCP evaluation in Semi-Live streams” and is the version v1.3.

The current version of this document aims at a detailed design of the modules presented in the Software Requirement Specification document of the on-going project.

The reader should have an idea about the various metrics involved in the performance measurement of TCP streams like RTT, Socket Setup Time and Data rate per stream. The glossary and abbreviations section gives the vocabulary of the terms used in the document and other technical information. The unit test plan for each module is covered in this document.

Release v1.4 on 2015-08-24

- In the document the tests which were repeated in the previous version is rectified.
- The string numbers of the tests are also updated according to the SRS and ATP.
- In section 6 of the document the information regarding RESTful API is updated.

Release v1.3 on 2015-06-01

- In section 3, Module 1: Frontend, the description is updated.
- In section 3.2, Unit Test Plan, all the tests i.e, M1_T1, M1_T2, M1_T3, M1_T4, M1_T5 have been changed and M1_T6 has been removed.
- In section 4.2, Unit Test Plan, all tests, M2_T1 and M2_T2 have been changed and M2_T3 has been added.
- In section 5.2, Unit Test Plan, all tests, M3_T1, M3_T2, M3_T3 and M3_T4 have been changed completely and the tests M3_T5 and M3_T6 have been removed.
- In section 6.1, Detailed Design has been updated.
- In section 6.2, Unit Test Plan, only the test M4_T1 has been changed and the rest of the tests have been removed completely.

Release v1.2 on 2015-05-20

- In all the sections of the document figure numbers and table numbers along with their captions are provided for the respective figures and tables.
- In section 3.2 of the document M1_T2 has been removed. M1_T3, M1_T4, M1_T5, M1_T6 has been added.
- In section 4.2, M1_T2 has been added.
- In section 5.1, the diagram has been updated.
- In section 6.2, M4_T2 has been updated.

Release v1.1 on 2015-05-14

- In section 3.1, figure 1 has been named. Figure 2 has been changed.
- In section 3.2, Unit test M1_T1 has been updated. Unit test M1_T3 has been added.
- In section 4.2, Unit test M2_T2: Scalability has been removed.

- In section 5.2, Unit test plans M3_T3, M3_T4, M3_T5 and M3_T6 have been added.

Release v1.0 on 2015-05-05

- Initial release

2. GLOSSARY AND ABBREVATIONS:

SRS - Software Requirement Specification

RRD – Round Robin Database
The data in RRD is mainly a time series data like network bandwidth, CPU load etc., which is stored in circular buffers.

DPMI – Distributed Passive Measurement Infrastructure
This structure allows for an efficient use of passive monitoring equipment in order to supply researchers and network managers with up-to-date and relevant data. [1]

API – Application Programming Interface
A set of routines, protocols and tools for building software applications.

RTT – Round Trip Time
The total time taken for a packet to be sent from source to destination and the acknowledgement received from destination to source.

GUI – Graphical User Interface
Allows the user to interact with electronic devices through graphical icons and audio-visual indicators.

TCP – Transmission Control Protocol
Protocol used to exchange data between two communicating hosts through an established connection.

3. MODULE 1: Frontend:

In this module, a WEB GUI is created with a simple dashboard. The tool will be provided with a simple user authentication. The user will be authenticated on the dashboard. The inputs connecting to the consumer are given on the dashboard. The required metrics will also be selectable. User will be able to define threshold levels for different metrics and also be able to provide an e-mail address for receiving threshold notifications on the dashboard. The above given inputs from the frontend are saved in the MySQL database.

The frontend will be able to display the output (evaluated data) as requested by the user. The evaluated data which is stored in database by the backend will be displayed on the webpage as requested. The frontend will be able to display the results in the visual-graphic format. The frontend is connected to RESTful API to connect with any third party.

3.1 Detailed Design:

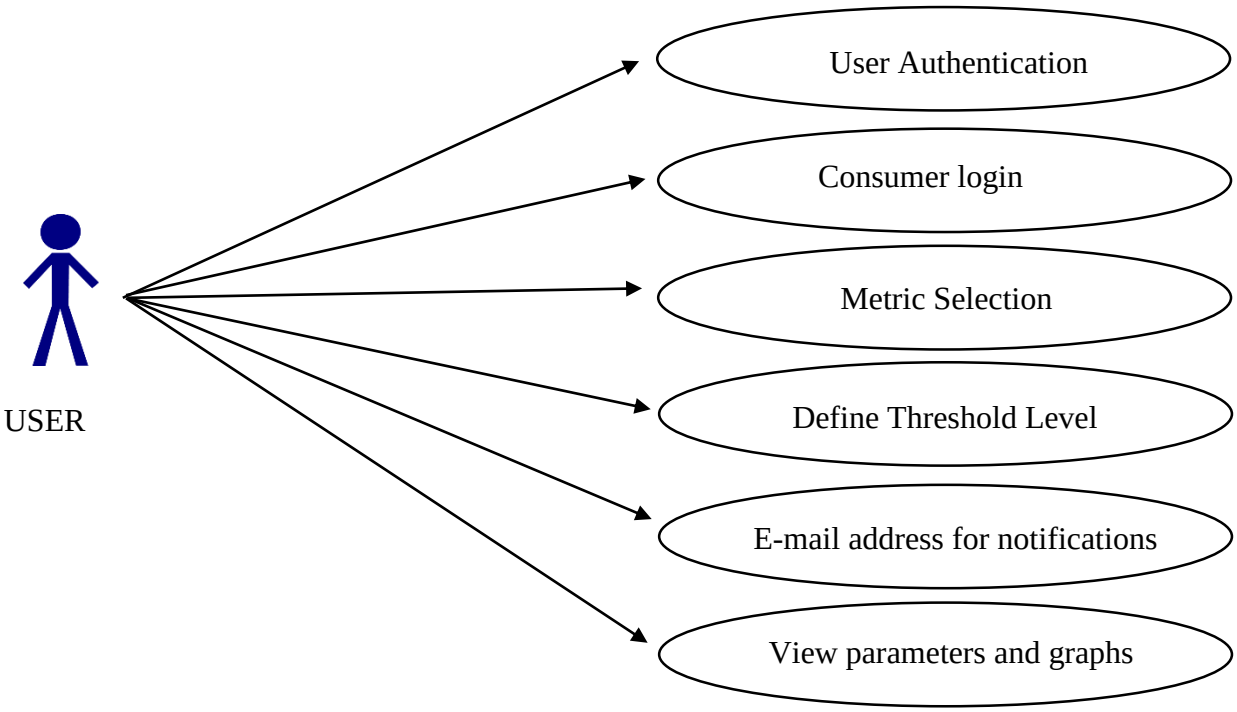

Fig 3.1: User Operations

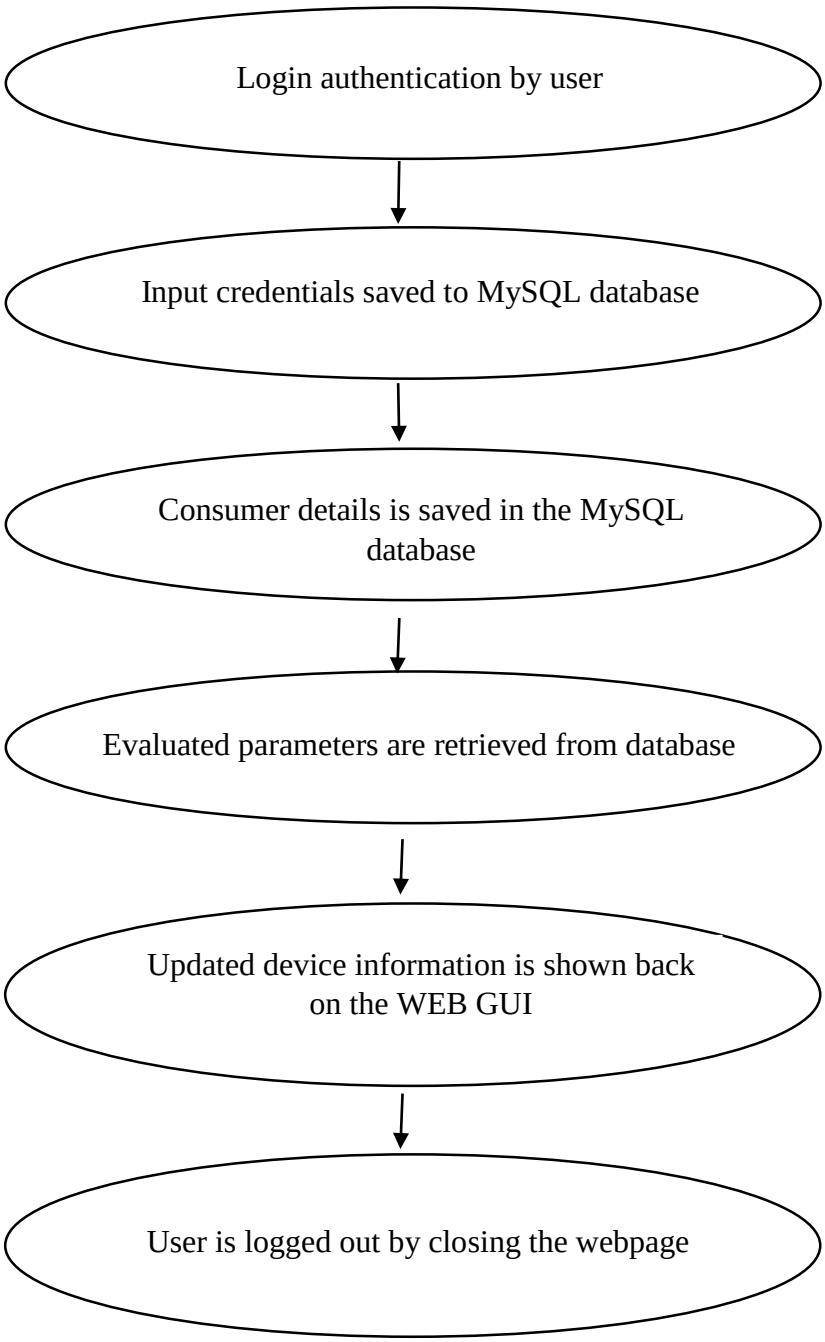

Fig 3.2: Frontend Process

The user will log on to the dashboard of the tool and then provide input credentials and stream identification which will be saved in the MySQL database. The evaluated parameters which will be saved in the database are displayed on webpage when requested. This process will be carried out on a continuous packet flow.

A URL is created and provided on the dashboard by which the user will be able to connect with the RESTful API and data can be exported. The webpage will also display the exported data in JSON format.

3.2 Unit Test Plan:

Test 1: Login Authentication Test

Test	ATP_1
Purpose	Login Authentication Test
Requirements	REQ_USRF_1
Environment	The webpage must be available for the user for accessing the tool.
Operation	• First, the user enters the username on the webpage of the tool. • Then the user enters the authentication password. • Next the user clicks on the login button.
Expected Result	The user will be authenticated into the tool.
Result	pass
Comment	

Table 3.1: ATP_1: Login Authentication test

Test 2: Metric Selection

Test	ATP_2
Purpose	Metric Selection
Requirements	REQ_USRF_7
Environment	The tool must be authenticated by the user.
Operation	• Firstly the user must log on to the webpage of the tool. • After logging in, the user will be directed to the dashboard of the tool. • The dashboard will contain the different options of the tool. • The user will be able to select the metric on the dashboard.
Expected Result	The user will be able to see the result regarding the selected metric of his choice on the dashboard.
Result	pass
Comment	

Table 3.2: ATP_2: Metric Selection

Test 3: Consumer login

Test	ATP_3
Purpose	Consumer login
Requirements	REQ_USRF_8
Environment	The tool must be authenticated by the user.
Operation	• First, the user must log on to the webpage of the tool. • The user will click on the consumer login option on the dashboard. • The user will then enter the details of the consumer on the webpage in the respective fields. • The user will next click on the submit button. • The user will now go back to the previous page and click on the any metric required.
Expected Result	The user will be able to see the result of the selected metric.
Result	pass
Comment	

Table 3.3: ATP_3: Consumer Login

4. MODULE 2: Database:

This module is used to store the data that appears during the operation of the tool. There are two types of databases namely MySQL and RRD. Data such as consumer details, user credentials etc., will be stored in the MySQL database. The time series data related to the evaluated parameters will be stored in the RRD database. RESTful API is also connected to the database so that it will be able to provide the data to any third party when requested.

4.1 Detailed Design:

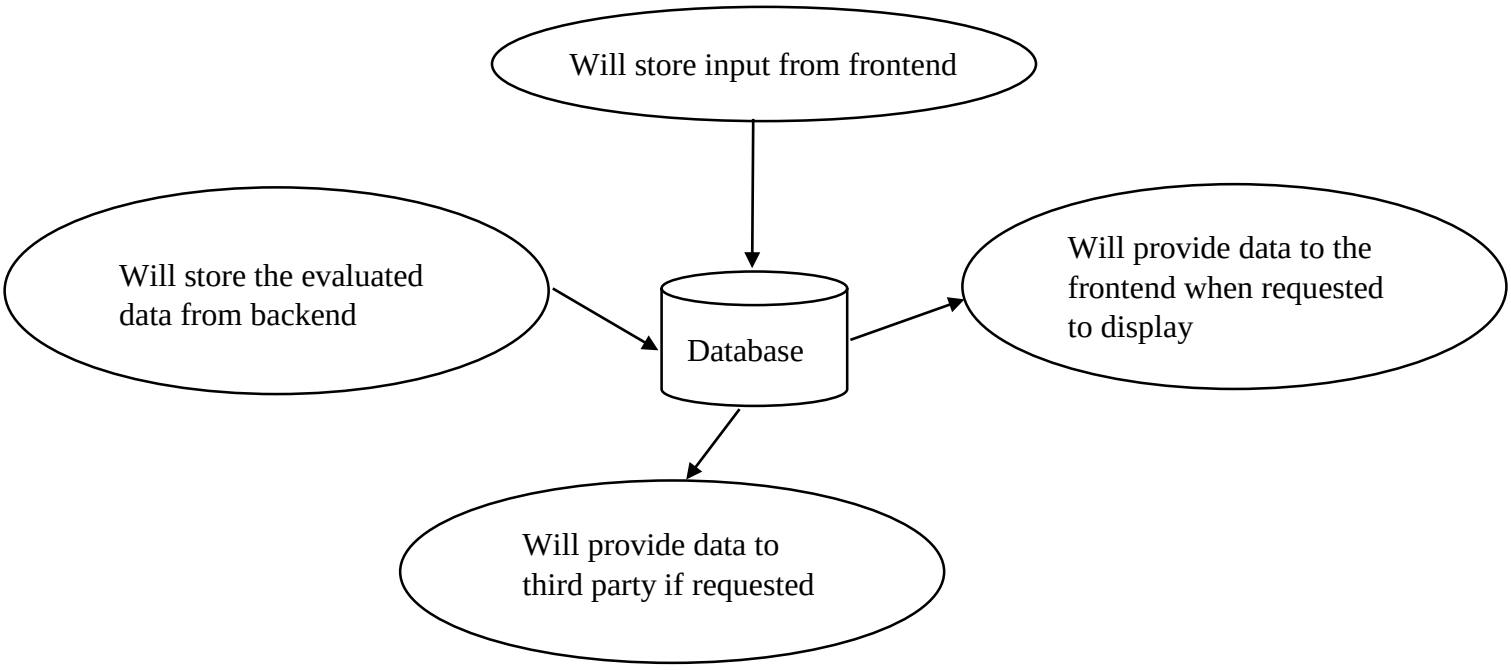

Fig 4.1: Processes of database

The database will store data gathered during the operation of the tool. It will store inputs like user credentials, consumer details etc., given at the frontend and provide the evaluated parametric data to the frontend to display as output. It will store the evaluated parametric data from the backend. It will also provide data to any third party through RESTful API when requested.

4.2 Unit Test Plan:

Test 4: Generation of graphs

Test	ATP_4
Purpose	Generation of graphs
Requirements	REQ_USRF_4
Environment	The connection between RRD and frontend must be established properly.
Operation	After the user logs on to the tool and the tool runs successfully, the user clicks on the “Statistics” button on the dashboard.
Expected Result	The graph will be displayed on the GUI.
Result	pass
Comment	

Table 4.1: ATP_4: Generation of Graphs

5. MODULE 3: Backend:

This module provides the parameters which are evaluated by the tool. The backend will capture the data packets and analyses it. The parameters such as RTT, socket setup time and data rate per stream are evaluated and stored in the database.

5.1 Detailed Design:

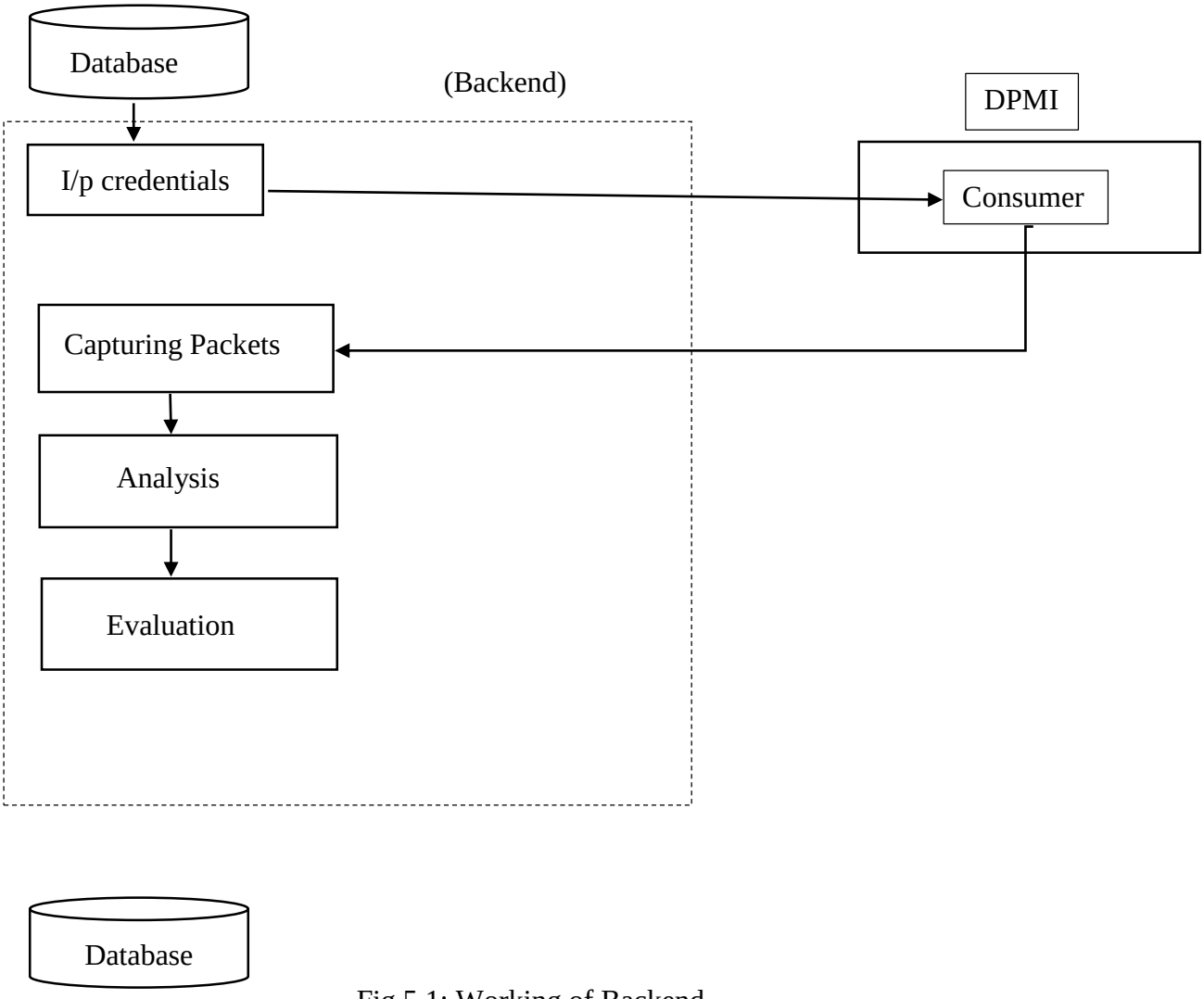

Fig 5.1: Working of Backend

Database will provide the required input credentials to the backend. The input credentials will contain the consumer details, hence a particular stream is captured from the consumer of DPMI. Then the captured stream will be analyzed and evaluated by the backend. The evaluated parameters namely RTT, Socket setup time and data rate per stream will then be stored in the database.

5.2 Unit Test Plan:

Test 5: Captured Packets are TCP packets

Test	ATP_5
Purpose	Captured Packets are TCP packets
Requirements	REQ_USRF_2
Environment	There should continuous flow of data stream.
Operation	• Open the terminal in the system with UBUNTU based operating system. • In the terminal the user firstly must connect with the consumer of DPMI. • The system needs to connect to the consumer of the network. This can be done by entering the username of the consumer in the network. ***ssh <username>@<IP address>*** • Then the password for connecting to the consumer will be asked. The user will be entering the password of the consumer. Password: ***password*** • Then the TCP packets are captured and stored in a ***cap*** file by using the following terminal command: ***sudo capdump -i eth2 -tcp -o <filename.cap> <stream>*** • The ***cap*** file is converted into a ***pcap*** file by using the following terminal command: ***cap2pcap -o <filename.pcap> <filename.cap>*** • Now the ***pcap*** file is run through the t-shark command using the following terminal command: ***tshark -r <filename.pcap> -O tcp***
Expected Result	The format of the captured packets is observed as TCP packets.
Result	pass
Comment	

Table 5.1: ATP_5: Captured packets are TCP packets

Test 6: Analysis of TCP packets

Test	ATP_6
Purpose	Analysis of TCP packets
Requirements	REQ_USRF_3
Environment	There should be a continuous flow of data.
Operation	• The user, after logging onto the dashboard, will enter the consumer details and then click on the "Submit" button. • Now the user will be selecting the required parameters on the dashboard like RTT, socket setup time or data rate per stream.
Expected Result	The requested parameters of TCP packets are displayed on WEB GUI
Result	pass
Comment	

Table 5.2: ATP_6: Analysis of TCP packets

Test 7: Threshold Notifications as e-Mails:

Test	ATP_7
Purpose	Threshold Notifications as e-Mails
Requirements	REQ_USRF_9
Environment	The e-mail address is provided by the user.
Operation	• The user must define the threshold levels on the dashboard. • The user must provide an e-mail address on the dashboard.
Expected Result	An e-mail is received with the notification to the given e-mail address.
Result	pass
Comment	

Table 5.3: ATP_7: Threshold notifications as e-Mails

Test 8: Threshold Notifications as traps:

Test	ATP_8
Purpose	Threshold Notifications as traps
Requirements	REQ_USRF_9
Environment	User must define a threshold level for the selected input.
Operation	• The user will define the threshold levels on the dashboard. • If the values exceed the threshold levels the traps will be sent. • Open the terminal and go into /var/log/syslog
Expected Result	Threshold notifications are seen as traps in the syslogs.
Result	pass
Comment	

Table 5.4: ATP_8: Threshold notifications as traps

6. MODULE 4: RESTful API:

The RESTful API will communicate between the tool and the third party. This will allow export of data to third party from the tool.

6.1 Detailed Design:

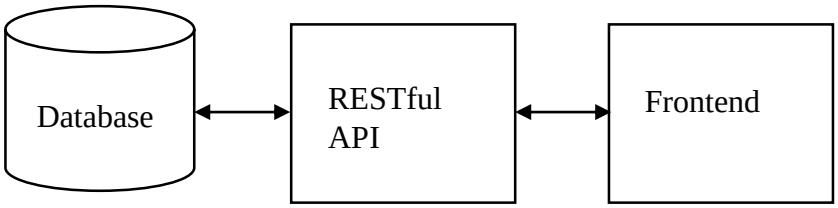

Fig 6.1: Usage of RESTful API

- The URL is created through which the third party gets connected with the tool.
- Connection between RESTful API and the database is established.
- Once the connection is established, the required tables in the database are accessed and converted into JSON format and exported through RESTful API. Also data is imported through RESTful API in JSON format and it is converted and saved in MySQL database.
- A RESTful interface of web service is designed with PHP script. It facilitates accessibility of data for a 3rd party through HTTP GET request and retrieves/store data from MySQL database shows service name, status and uptime in JSON format.
- The data in the arrays, which is in JSON format, is echoed on the screen.

The RESTful API will allow the tool to export the data to any third party as requested. And also will be able to import the data from any 3rd party data source.

6.2 Unit Test Plan:

Test 9: Data accessibility through RESTful API

Test	ATP_9
Purpose	Data accessibility through RESTful API
Requirements	REQ_USRF_5, REQ_USRF_6
Environment	The link for the connection to the RESTful API
Operation	• The user will click on the given URL. • A new webpage appears with the imported data in JSON format.
Expected Result	The imported data is seen on the webpage.
Result	pass
Comment	

Table 6.1: ATP_9: Data accessibility through RESTful API

7. References:

[1] Patrik Arlos, Markus Fiedler, and Arne A. Nilsson. *A Distributed Passive Measurement Infrastructure,* In Passive and Active Measurement Workshop (PAM05), US, 2005.

[2] Ian Sommerville, *SOFTWARE ENGINEERING*, 9th ed. Pearson Publications, 2011.