

INTEGRATION OF A WEB PORTAL AND AN ERP THROUGH WEB SERVICE BASED IMPLEMENTATION AND TESTING APPROACH

Manishkumar Solanki¹, Seema Shah², Deepali Vora³

¹M.E. Student, Information Technology, Vidyalkar Institute of Technology, Mumbai, India

²Assistant Professor, Information Technology, Vidyalkar Institute of Technology, Mumbai, India

³Assistant Professor, Information Technology, Vidyalkar Institute of Technology, Mumbai, India

Abstract

The present business world is facing challenges in accessing their fragmented data to create an accurate and consistent view of their core information assets across the enterprise to drive business decisions and operations. To deliver timely, trustworthy and holistic information to various stakeholders, the enterprise has to employ promising data integration architecture. The Web Services approach based on service oriented architecture provides business agility through loose coupling and reusability of data assets. It enables applications and business processes to access relevant data consistently and accurately whenever it is needed no matter wherever and in whichever form it resides. This paper implements SOAP based web services to integrate data between a web portal and an ERP system. The Performance Testing is also conducted to study the behavior of web services under varying loads.

Keywords: Web Services, SOAP, iText, JFreeChart, Axis2, Apache POI

-----***-----

1. INTRODUCTION

Organizations today are facing huge data challenges, especially in this fast-changing tech world where assessing, understanding and leveraging data drive the competitive edge. Along with well established business model one has to deliver improved service, comply with increasing government and industry regulations and standards, keep costs low, and generate more revenue. [1] To fulfill the requirements of different stakeholders, an enterprise has to generate a unified view of various business processes which are implemented through diverse but coexisting applications. These applications are developed in different languages and are running on heterogeneous platforms. A smart and reliable data integration strategy is to be devised to gain a more comprehensive view to satisfy the needs by combining backend complementing business applications. Service Oriented Architecture (SOA) has given a new dimension in the data integration arena by offering qualities such as business agility, reusability, availability, scalability and testability of applications. Web Service, a message driven architecture realizes the concept of SOA. The use of web services eliminates the issues of application servers, operating systems, protocols, or devices across the enterprises.

This paper implements SOAP based Web Services to integrate data between a web portal and an ERP system. The services are also tested to observe their response times under varying loads. The remainder of the paper is organized as follows: Section 2 describes the identified web services. Section 3 discusses the core technologies needed to integrate a web portal and an ERP system. Section 4 illustrates the implementation architecture of web services. Section 5 shows the outcomes of web services. Section 6 discusses the

results of performance testing of web services and finally Section 7 concludes the paper.

2. DESCRIPTION OF IDENTIFIED WEB SERVICES

Vidyalkar Institute of Technology (VIT), Wadala (E), Mumbai has implemented an academic and social networking web portal to satisfy the requirements of its' prime stakeholders i.e. students, faculties and management. The requirements of these stakeholders got fulfilled with the foundation of VIOLA which stands for 'Vidyalkar Online Applications'.

VIT is also running an ERP System named VERP which integrates and automates the data of Accounts, Admission, Exam, Library, Human Resources, Administration, Management, Payroll and other departments on a single computer system providing a unified central access of information of entire institute. [2]The requirements and system architecture to integrate both the systems are given in our paper [2]. In this paper, we have implemented web services for the following requirements:

- The student details from VERP should be integrated to VIOLA to generate the Bonafide Certificate and to send it to the student through an email.
- The Company and the student details from VERP should be integrated to VIOLA so that the Training and Placement Officer can conduct the campus interview online.

To satisfy the above requirements we have developed the following web services:

1. **Bonafide Service:** This service generates a Bonafide Certificate in pdf format for the requesting student.
2. **SendBonafide Service:** This service sends generated Bonafide Certificate (pdf document) as an email to the requesting student.
3. **Campus Service:** The TPO conducts campus through an html form and invokes this web service. This service will search the eligible students who fulfill the criteria of a selected company and sends email notification to them. It also updates the status of these students in the database.

The email notification briefs the student about campus interview and specifies last date of submitting their interest for the campus. To show willingness or not, the student has to click YES/NO hyperlinks.

4. **Campus Status Service:** This service describes student selection proportion in graphical form. The TPO can review the campus selection statistics i.e. the no. of students appeared, no. of students cleared aptitude, cleared G.D., Personal Interview and finally absorbed by the company for the present year, for the last three years or for all the companies in the present year in the form of bar charts. It also provides overall selection proportion of all companies for a particular year through Pie Charts.
5. **Campus Students List Service:** This service generates various students list i.e. invited, interested and selected in the form of Excel sheets. The TPO can give the choice to get the type of list he is interested in. It also generates a consolidated list of all selected students in the present campus year along with their respective details.

To realize above web services we require some technologies to be installed at VIOLA and VERP side. These technologies are described in the next section.

3. CORE TECHNOLOGIES

Technologies provide the backbone for the realization of an implementation from the abstract ideas. In a software oriented project, along with computer system, we require some application softwares to make it a reality. Following technologies will be used in the implementation of VIOLA-VERP integration Web Services.

3.1 Technologies to be installed on VIOLA Server:

The student and TPO are using VIOLA GUI interface to get their functionalities to be done. These users are under the impression that they are receiving their responses from the VIOLA only. In the reality the VIOLA invokes the Web Services situated at the VERP to get done the requirements of these users. The VIOLA uses following technologies:

- a. **PHP:** It is a server-side scripting language used to generate dynamic web pages. VIOLA is developed with PHP to generate dynamic responses.
- b. **MySQL:** It is an open source, Enterprise-level, multi-threaded, relational database management system. VIOLA databases are stored on this DBMS.
- c. **Apache Server:** Apache server is an application server for PHP. It acts as a container for PHP scripts. Apache server resides at the server end to process the server side PHP scripts and delivers HTML content to the client. It uses HTTP protocol to listen client requests.
- d. **XAMPP Server:** XAMPP stands for Cross-Platform (X), Apache (A), MySQL (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing purposes. It also comes with phpMyAdmin, which provides GUI environment to work with MySQL database. XAMPP also supports SOAP based client as well as server functionalities. The VIOLA uses SOAP Client functionality to invoke the web services.

3.2 Technologies to be installed on VERP Server:

The web services will be deployed either on the third server as per shown in System Design diagram [2] or on the VERP server. We have deployed these services on the VERP server. To achieve this we have to install other softwares / APIs which are enlisted as below:

- a. **Java:** It is an object oriented and platform independent programming language. Web Services will be developed in Java.
- b. **Apache Tomcat Server:** It is an open source web server and servlet container developed by the Apache Software Foundation (ASF). It provides a "pure Java" HTTP web server environment for Java code to run in. Apache Tomcat also provides run time environment for the Axis toolkit which is used to develop SOAP based web services. Web Services are deployed inside the tomcat environment.
- c. **Apache Axis 2:** It is a third-generation open source SOAP stack used to deploy Web Services.[3]
- d. **iText:** It is a free, open source library that lets us create and deal with PDF files with Java code. It offers support for a wide range of advanced PDF features, including digital signatures, forms, and encryption.[4]
- e. **Apache POI:** It is a project run by the Apache Software Foundation, provides pure Java libraries for reading and writing files in Microsoft Office formats, such as Word, PowerPoint and Excel. [5]
- f. **JFreeChart:** It is an open-source framework for Java, which allows the creation of a wide variety of both interactive and non-interactive charts such as X-Y charts, bar charts, pie charts, specific charts, etc.[6]

The installation of above technologies with appropriate path and classpath setting as well as port configuration in case of web servers at both VIOLA and VERP ends sets up the environment ready for the data integration. The next section illustrates the implementation architectures for the said web services.

4. IMPLEMENTATION ARCHITECTURE

The architecture of a software system consists of its structure, decomposition into components, and their interfaces and relationships. The implementation architecture shows the flow of invocation of web services from the client (VIOLA) to the server (VERP) and the reply from the server in the form of requests and responses.

Fig.1 depicts the invocation process of Bonafide Service. The student submits a request to get Bonafide certificate by filling the form which asks the roll number of that student. If the input is valid then the student receives his/her bonafide certificate in PDF format via Email otherwise it will get the message that he/she is not a valid student of VIT.

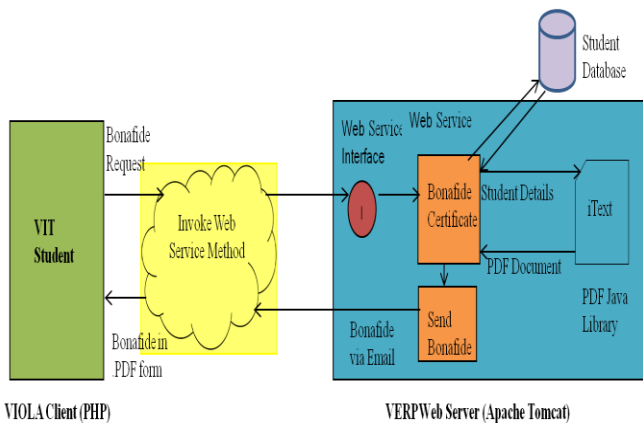


Fig -1: Student bonafide generation Web Service Architecture

Fig. 2 depicts how the VIOLA server invokes various web services to conduct the campus activity smoothly for the TPO. When he selects campus conduction operation, ConductCampus Web Service is being called and upon successful operation it sends the status message to the TPO. CampusStatusChart service is invoked when TPO selects either year wise or company wise option to get the campus statistics charts. In response, he gets a web page displaying the bar chart or the pie chart according to the selected option. To generate the list of invited, interested and selected students, TPO has to select appropriate option and he gets the respective list in Excel sheet format.

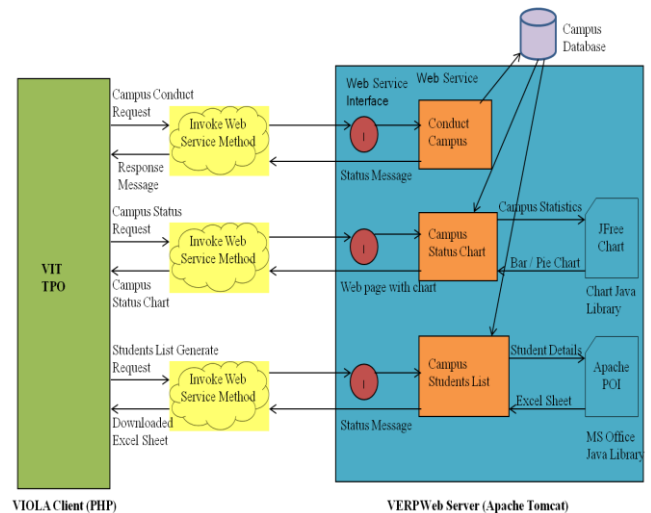


Fig -2: TPO related Web Services Architecture

On the basis of above architectures, the web services have been implemented.

The Flow Chart drawn in Fig. 3 shows the flow of execution of web services. The VIOLA provides a GUI interface (web page) to the user. Users are of two types (i) student and (ii) TPO. If the user is a student then he can request for the bonafide certificate and if the user is a TPO he can conduct campus related activities.

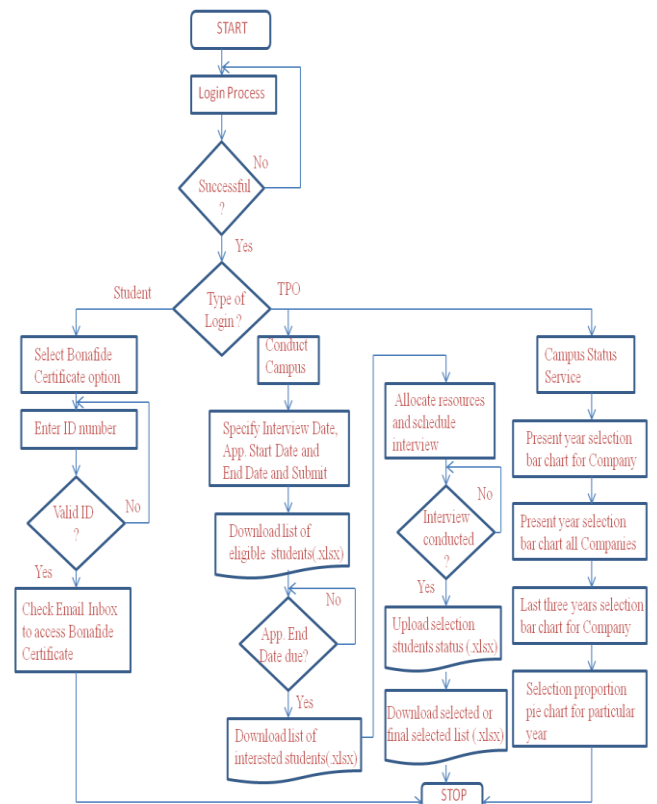


Fig -3: Execution Flow of VIOLA-VERP Integration System

The results of the implementation are discussed in the next section.

5. RESULTS OF WEB SERVICES

The result section represents the research findings and analysis of those findings. The aim of the discussion of results is to outline and explain the results and to relate them to others' theories and results. It also conducts evaluation of the methodology of the experiment and provides suggestions for further research. The outcomes of implemented web services are explained in this section in the form of module implementation.

5.1 Login Page

The entry point for the VIOLA-ERP integration system login form which is shown in Fig. 4. The user is having choice to login as a student, TPO.

Fig -4: Login Form for VIOLA-VERP Integration System

5.2 The Student

After login, the student has to click on the Bonafide tab. The student gets the form as shown in Fig. 5. He has to provide his Id number and has to submit the form to receive the bonafide certificate.

Fig -5: Bonafide Certificate request Form

The student will receive the bonafide certificate immediately in his mail box. The format of bonafide certificates is shown in Fig. 6. If the student is not a valid student then he will get the appropriate message.

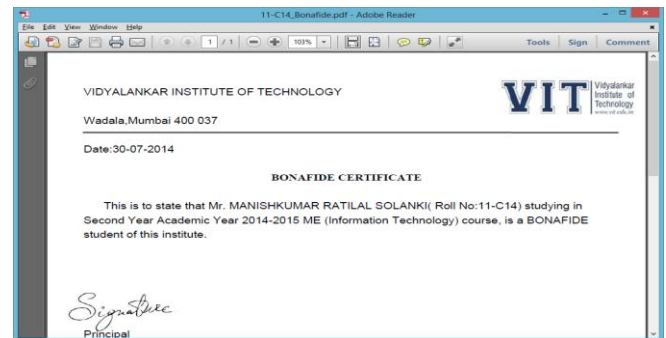


Fig – 6: Bonafide Certificate Format

5.3 The TPO

After login, the TPO can conduct various campus activities i.e. adding a new company, updating a company details, campus conduction, chart generation, and generation of list of campus students. These modules are described as below.

5.3.1 Campus Conduction

The TPO can conduct campus by pressing “Conduct Campus” menu item under “Campus” menu as shown in Fig. 7. The TPO will select one of the companies and date of interview as well as the Application Start Date and Application End Date which represents the range of date for the students to show their interest for the interview.

Fig -7: Campus Notification Form

As the TPO submits the form, the “CampusService” web service is invoked by the form which sends emails to the eligible students who satisfy the criteria of the selected company. The mail box which received campus notification message is shown in Fig. 8.

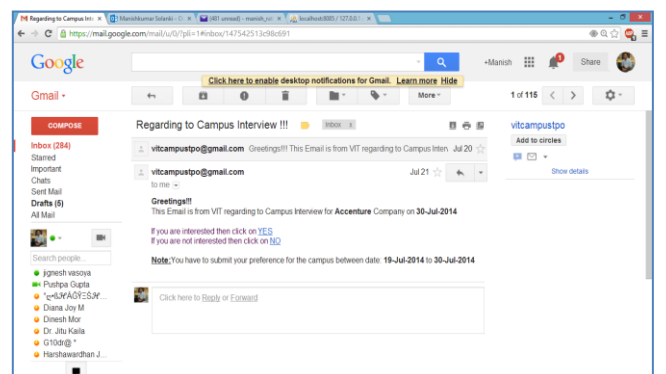


Fig – 8: Email sent to Gmail user for the campus interview in Accenture

The students have to submit the status of their interest in the campus interview by pressing Yes or No options within specified dates given in the email content.

5.3.2 Campus Charts

The TPO has to fill the form for the campus statistics after conduction of the campus every year. The campus charts represent the performance of students in the campus in graphical form with the help of bar and pie charts. Fig. 9 shows the campus selection proportion of Infosys company for the recent year. Fig. 10 shows selection proportion of all companies for the present year. Fig. 11 shows the selection proportion of Accenture for last three years. Whereas Fig. 12 shows the selection proportion of students in all companies for the present year in the form of pie chart

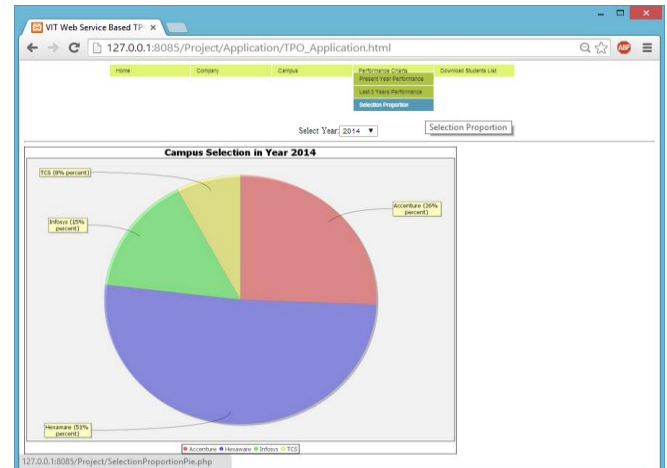


Fig -12: Campus Statistics of all companies for year 2014

5.3.3 Campus Students List

The campus lists represent various lists such as invited (eligible), interested and selected students. The invited list represents the students who are eligible for the particular company. The campus student’s list form provides two types of choices i.e. “Consolidated” list as shown in the Fig. 13 which gives the list of students of all disciplines and “Divisionwise” list as shown Fig. 14 which gives the list of students division wise. These options are provided for all three types of category i.e. invited, interested and selected.

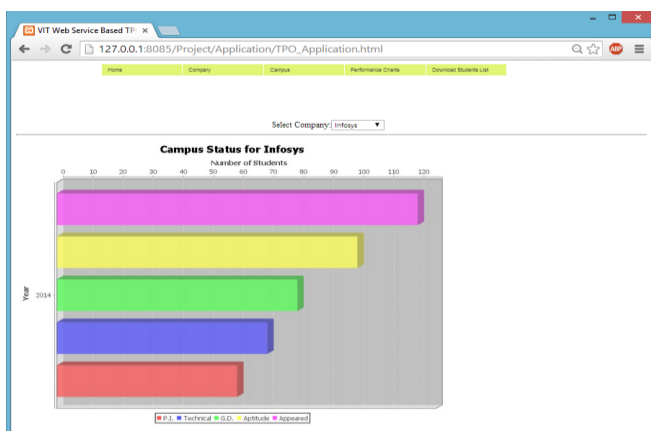


Fig – 9: Campus Statistics of Infosys for year 2014

| Sr.No. | ID | Invite_ID | LName | FName | MName | Branch |
|--------|-----|-----------|---------|-------------|---------|----------|
| 1 | 103 | 51 | Patil | Manish | Namdev | CMPN-II |
| 2 | 107 | 52 | Solanki | Manishkumar | Ratilal | INFT-I |
| 3 | 108 | 53 | Goradia | Jay | S | INFT-I |
| 4 | 110 | 54 | Doshi | Tosha | N | INFT-II |
| 5 | 111 | 55 | Doshi | Viral | M | CMPN-III |
| 6 | 112 | 56 | Dodia | Rohan | L | INFT-II |
| 7 | 113 | 57 | Joshi | Raj | B | CMPN-II |

Fig -13: Consolidated list of invited students for campus of Accenture

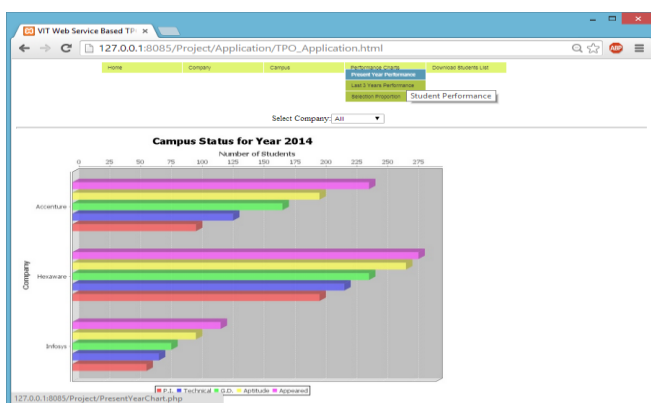


Fig -10: Campus Statistics of all companies for year 2014

| Sr.No. | ID | Invite_ID | LName | FName | MName | Branch | company_name |
|--------|-----|-----------|---------|-------------|---------|--------|--------------|
| 1 | 107 | 52 | Solanki | Manishkumar | Ratilal | INFT-I | Accenture |
| 2 | 108 | 53 | Goradia | Jay | S | INFT-I | Accenture |

Fig -14: Division wise list of invited students for campus of Accenture

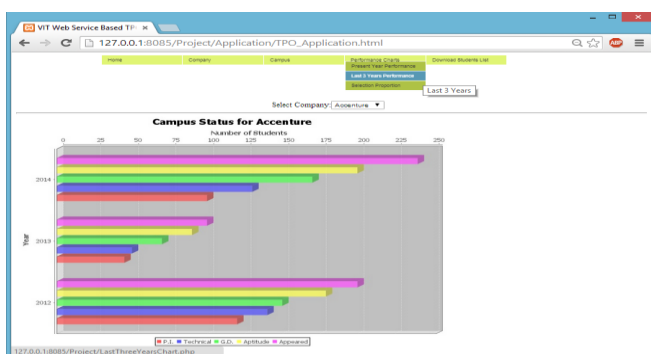


Fig – 11: Campus Statistics of Accenture for last 3 years

After receiving an email, the student has to click on Yes hyperlink if he is interested in the campus and if he is not interested in the campus he has to press No. The database will be modified according to given preference at the server end. The list of interested students is shown in Fig. 15. The student named “Doshi Tosha N” has not shown interest in the campus so her name is not enlisted in the interested students list.

| Sr.No. | ID | Invite ID | LName | FName | MName | Branch |
|--------|-----|-----------|---------|-------------|---------|----------|
| 1 | 103 | 51 | Patil | Manish | Namdev | CMPN-II |
| 2 | 107 | 52 | Solanki | Manishkumar | Ratilal | INFT-I |
| 3 | 108 | 53 | Goradia | Jay | S | INFT-I |
| 4 | 111 | 55 | Doshi | Viral | M | CMPN-III |
| 5 | 112 | 56 | Dodia | Rohan | L | INFT-II |
| 6 | 113 | 57 | Joshi | Raj | B | CMPN-II |

Fig – 15: List of interested students for campus of Accenture

After conducting campus interview, the TPO will prepare a list with one extra column named “selection” which gives the status of the student whether he/she is selected or not. If he/she is selected, the status will be Yes. otherwise it will be No. This is shown in the Fig.16. This list will be uploaded by the TPO to the VERP through the web service.

| Sr.No. | ID | Invite ID | LName | FName | MName | Branch | Selection |
|--------|-----|-----------|---------|-------------|---------|---------|-----------|
| 1 | 103 | 51 | Patil | Manish | Namdev | CMPN-II | No |
| 2 | 107 | 52 | Solanki | Manishkumar | Ratilal | INFT-I | Yes |
| 3 | 108 | 53 | Goradia | Jay | S | INFT-I | Yes |
| 4 | 111 | 55 | Doshi | Viral | M | CMPN-II | Yes |
| 5 | 112 | 56 | Dodia | Rohan | L | INFT-II | Yes |
| 6 | 113 | 57 | Joshi | Raj | B | CMPN-II | Yes |

Fig – 16: Selection status list for campus of Accenture

The TPO will upload selection status list to update the database. After uploading the selection status file, TPO can generate list of company wise selected students as shown in Fig. 17 as well as after the overall campus activity ends the list of final selected students’ list as shown in Fig.18.

| Sr.No. | ID | Invite ID | LName | FName | MName | Branch |
|--------|-----|-----------|---------|-------------|---------|----------|
| 1 | 107 | 52 | Solanki | Manishkumar | Ratilal | INFT-I |
| 2 | 108 | 53 | Goradia | Jay | S | INFT-I |
| 3 | 111 | 55 | Doshi | Viral | M | CMPN-III |
| 4 | 112 | 56 | Dodia | Rohan | L | INFT-II |
| 5 | 113 | 57 | Joshi | Raj | B | CMPN-II |

Fig – 17: The list of selected students for Accenture

| Sr.No. | ID | LName | FName | MName | Branch | company_name | Intv Date | package |
|--------|-----|---------|-------------|---------|----------|--------------|------------|---------|
| 1 | 107 | Solanki | Manishkumar | Ratilal | INFT-I | Infosys | 31/07/2014 | 4.7 |
| 2 | 107 | Solanki | Manishkumar | Ratilal | INFT-I | Accenture | 30/07/2014 | 3.5 |
| 3 | 108 | Goradia | Jay | S | INFT-I | Accenture | 30/07/2014 | 3.5 |
| 4 | 111 | Doshi | Viral | M | CMPN-III | Accenture | 30/07/2014 | 3.5 |
| 5 | 112 | Dodia | Rohan | L | INFT-II | Accenture | 30/07/2014 | 3.5 |
| 6 | 113 | Joshi | Raj | B | CMPN-II | Accenture | 30/07/2014 | 3.5 |

Fig – 18: The Final Selection List

6. RESULTS OF PERFORMANCE TESTING OF WEB SERVICES

Software testing helps to make sure that it meets all the requirements it was supposed to meet by pointing out the errors, if any, while using the software. As Web Services are loosely coupled and autonomous components which are individual units of business logic in SOA, a distinguished approach for testing the Web Services is employed. This gives the ability to testers to follow a component level testing methodology.

A web service can be tested to verify both functional and non-functional requirements. We have tested our web services with SOAP UI Pro testing tool which an industry proven testing tool. The implemented web services have been tested with both functional and non-functional approaches. They are described as below:

6.1 Functional Testing of Web Services:

The primary objective of the functional testing is to achieve expected business functionality from the system. This process can be done manually by running the system and giving different types of permissible and non permissible values. Since the web service interface is a machine readable XML document, it is not an easy task to read the WSDL and derive tests manually. The assertions (testing criteria) selected for the Bonafide Service in SoapUI Pro 5.1.1 are given in Table 1.

Table -1 List of applied assertions on BonafideService

| No. | Assertion | Function | Value |
|-----|--|---|--|
| 1 | SOAP Response | The response must be SOAP response | NA |
| 2 | Response SLA (Service Level Agreement) | The maximum time (ms) within which the response must come | 30ms |
| 3 | Not Contains | The response should not contain this value | You are not a Student of this institute. |
| 4 | Not SOAP Fault | The response must not be a Fault response | NA |
| 5 | Valid HTTP Status Codes | HTTP status codes in the response must be within the specified status codes in this assertion | 200 |

The result of the functional test is shown in Fig.19. As the request argument is 11-C14 (Student ID), the returned response message is “Bonafide has been successfully generated”. As the response message satisfies the conditions of all the assertions, they are shown as VALID state in green color.

Similarly, we have tested other web services and they are satisfying the business logic expectations.

6.2 Performance Testing of Web Services:

Performance Testing is conducted to test the behavior of the system in a given workload. The outcomes of performance testing give us the quality measures such as reliability, scalability, consistency, throughput, response time, etc. Due to use of relatively complex XML message processing and promise of communication with heterogeneous systems, there is a lot of heavy XML serialization/de-serialization tasks used. These complexities multiply by greater levels when the messages are enriched with various Quality of Service (QoS) options such as the WS-* headers i.e. encryption.[7] With these facts, there can be a considerable slowness introduced by SOAP web services. In most cases, when we publish a web service for consumers, we are expected to define a valid Service-level Agreement (SLA). Hence, it is important to do performance or scalability tests with individual web services beforehand and verify whether the published SLA figures are realistic and achievable.

The different Load Strategies available in SoapUI Pro allow us to simulate various types of load over time, enabling us easily test the performance of our target services under a number of conditions. SoapUI Pro also allows users to configure various load testing options such as introducing delays in between threads to simulate real world use cases and run tests in burst mode to test stress of services. The various performance testing strategies provided by the SoapUI Pro are described in Table 2 along with their functionality and applications.

Table –2: Description of Performance Testing Strategies

| No. | Strategy | Function | Application |
|-----|------------|--|---------------------------------|
| 1 | Simple | Runs the specified number of threads (virtual users) with the specified delay between each run to simulate a "breathing space" for the server | Load Testing |
| 2 | Fixed Rate | Guarantees a number of executions (tps) within certain time-throughput testing | Baseline, Load and Soak Testing |
| 3 | Variance | Varies the number of threads over a time in a "saw tooth" manor. Configured by setting the Interval to the desired value and the Variance to the how much the number of threads should decrease and increase | Variable Load Testing |
| 4 | Burst | Does nothing for the configured Burst Delay, then runs the | Recovery Testing |

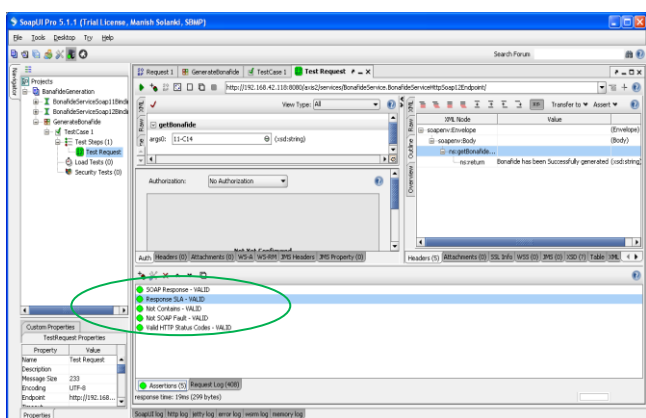


Fig –19: Test Result of valid assertions applied for BonafideService

If the student provides wrong ID or the student has been passed out from the college then the assertion “Not Contains” will become FAIL which is shown in Fig. 20 with red color.

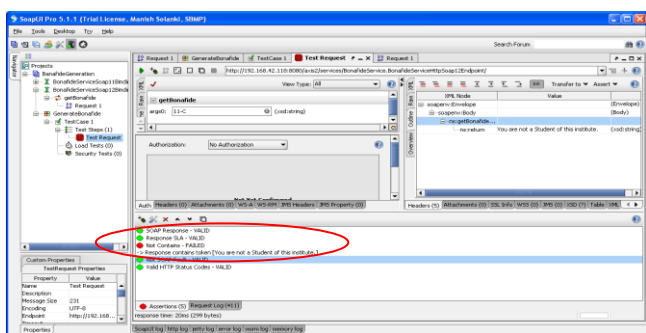


Fig –20: Test Result of Failed assertions applied for BonafideService

| | | | |
|---|--------|---|---|
| | | configured number of threads for the “Burst Duration” and goes back to sleep. | |
| 5 | Thread | Linearly changes the number of threads from one level to another over the run of the Load Test | To identify the moment of failure |
| 6 | Grid | Allows us to configure the relative change in number of threads over time. | To see service behaviour under varying load |
| 7 | Script | The script we specify is called regularly and should return the desired number of threads at that current time. | Custom Load Testing |

The above strategies are applied for all the implemented web services. In this paper we have discussed mainly two web services (1) Bonafide Service and (2) Campus Status Service and (3) Campus Students List Service.

6.2.1 Bonafide Service Performance Testing:

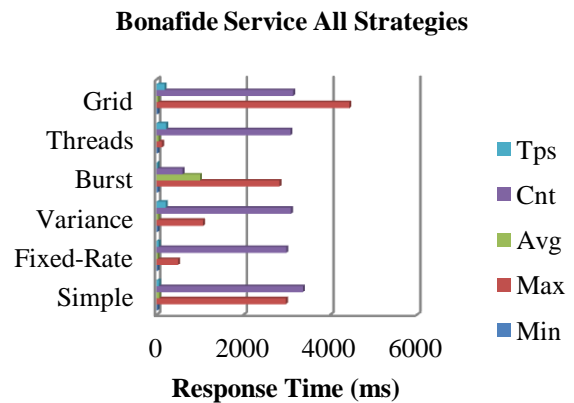
The parameters selected for the Bonafide Service are given below:

- **Simple:** Threads=60 (Stress Testing), random=0, Test Delay =1000ms
- **Fixed- Rate:** Rate=50, max. threads= 20 and Limit=60 seconds
- **Variance:** Threads=8, Interval=20, Variance=0.2 and Limit=20 seconds
- **Burst:** Threads=50, Burst Delay=10ms, Burst Duration=10ms and Limit=60 seconds
- **Thread:** Start Threads=1, End Threads=10 and Limit= 60 seconds
- **Grid:** Starts with 1 thread and then time(seconds) and threads are given as 12 - 5, 24 - 1, 36 - 6 and 48 - 4

The statistics of all the above tests are shown in Table 3 and column chart is shown in Graph 1.

Table -3: Statistics of all Strategies for Bonafide Service

| Strategy | Min (ms) | Max (ms) | Avg (ms) | Cnt | Tps |
|------------|----------|----------|----------|------|--------|
| Simple | 22 | 2984 | 57.87 | 3375 | 56.22 |
| Fixed-Rate | 21 | 495 | 24.77 | 2991 | 49.85 |
| Variance | 23 | 1071 | 36.30 | 3099 | 212.91 |
| Burst | 30 | 2832 | 1001.02 | 598 | 22.75 |
| Threads | 21 | 127 | 43.42 | 3085 | 220.26 |
| Grid | 21 | 4443 | 31.58 | 3149 | 186.19 |



Graph –1: Performance results of all strategies for Bonafide Service

As this service is retrieving data from the database (at remote machine) and generating .PDF document, it requires some time to settle down after generating the document. Due to this reason if we increase the number of threads after certain limit, server starts giving “Exceptions” which increases the average response time (Avg) value whereas the server is not giving any output. The above graph shows higher value of Avg parameter for the Burst Strategy because in this strategy, all the threads are attacking (requesting) simultaneously on the server for the specified Burst Duration.

6.2.2 Campus Status Service Performance Testing:

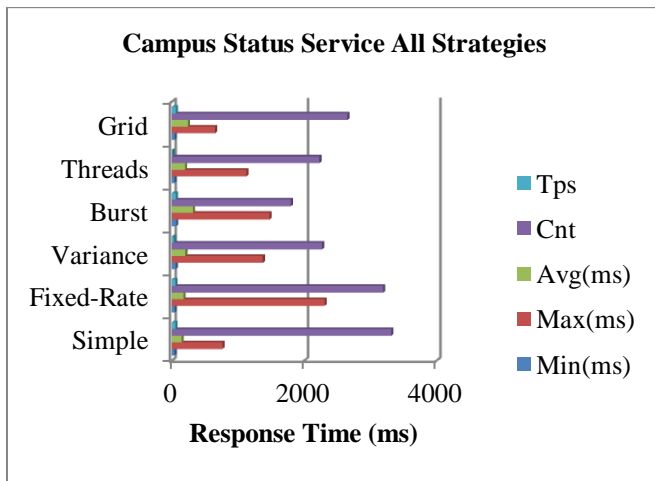
The parameters selected for the Campus Status Service are given below:

- **Simple:**Threads=50 (Stress Testing), Random=0.5, Test Delay =1000ms
- **Fixed- Rate:** Threads=39, Rate=55, Max. Threads=100 and Limit=60 seconds
- **Variance:** Threads=20, Interval=60, Variance=0.5 and Limit=60 seconds
- **Burst:** Threads=20, Burst Delay=10ms, Burst Duration=10ms and Limit=60 seconds
- **Thread:** Start Threads=1, End Threads=50 and Limit= 60 seconds
- **Grid:** starts with 2 threads and then time(seconds) and threads are given as 12 - 1, 24 - 5, 36 - 10 and 48 - 15

The statistics of all the above tests are shown in Table 4 and column chart is shown in Graph 2.

Table –4: Campus Status Service with all Strategies

| Strategy | Min (ms) | Max (ms) | Avg (ms) | Cnt | Tps |
|------------|----------|----------|----------|------|-------|
| Simple | 41 | 766 | 148.32 | 3317 | 55.24 |
| Fixed-Rate | 44 | 2309 | 178.45 | 3193 | 53.21 |
| Variance | 59 | 1376 | 204.94 | 2270 | 34.62 |
| Burst | 65 | 1477 | 319.89 | 1799 | 60.77 |
| Threads | 41 | 1128 | 198.75 | 2231 | 12.34 |
| Grid | 44 | 653 | 243.53 | 2653 | 60.98 |



Graph – 2: Performance results of all strategies for Campus Status Service

This service requires less settling time compared to the Bonafide certificate because it creates a .jpg file to display the bar or pie chart. As it requires less processing time, we can achieve much throughput i.e. higher rate of Tps. The Avg value of Burst Strategy is higher than other strategies as discussed in Bonafide Service.

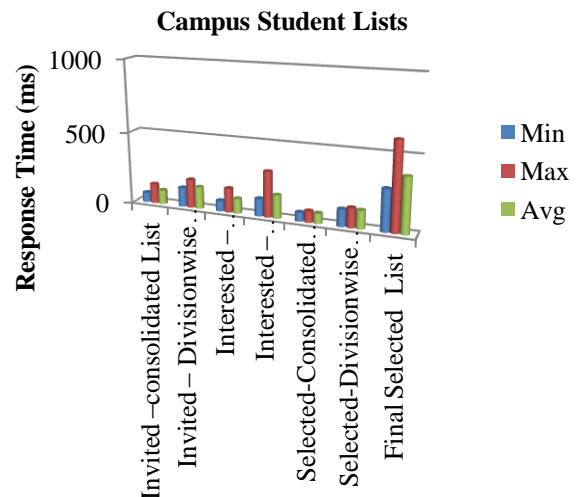
6.2.3 Campus Students List Service Performance

Testing:

The Campus Student List Service is made of some small web services which generate various lists for the TPO. As these web services are being invoked by the TPO only (single user) they are not tested with the testing strategies. They are tested response time point of view only. Table 5 shows the statistics of these web services.

Table – 5: Response times of Campus Students List Services

| Web Service | Min (ms) | Max (ms) | Avg (ms) |
|--------------------------------|----------|----------|----------|
| Invited –consolidated List | 67 | 135 | 97.29 |
| Invited – Divisionwise List | 132 | 196 | 150.38 |
| Interested –Consolidated List | 73 | 165 | 101.08 |
| Interested – Divisionwise List | 120 | 313 | 159.13 |
| Selected-Consolidated List | 60 | 77 | 69.25 |
| Selected-Divisionwise List | 117 | 133 | 124.30 |
| Final Selected List | 284 | 600 | 375.65 |



Graph –3: Performance results of all strategies for Campus Status Service

The statistics show that the generation of Divisionwise list of students takes much time than the consolidated list in case of Invited, Interested and Selected lists. The Final Selected List service takes much time compared to other services because it generates a list of all the students who are selected in various companies in particular time duration i.e. one month. (Campus Month) along with their packages.

7. CONCLUSION

The Integration of VIOLA and VERP System through SOAP has been implemented by developing various SOAP based Web Services. Web Services provide loose coupling and platform as well as language independency compared to the traditional web based applications where platform plays a vital role in the selection of application development language. The advantage of this system is that it can be deployed on any operating system as the Web Services are developed in Java, serving the requests of a web portal developed in PHP. There will be no change or modification in the Web Services even though we change the client. Small changes will be made to invoke the web services at the client side depending on the language in which the client has been developed. Along with this, as Web Services are developed in modular fashion, the task of debugging and fault finding becomes so much easier

The Integration of VIOLA and VERP System through SOAP is tested with SoapUI Pro, an industry proven testing tool to ensure the functional quality metrics such as usability reliability, consistency, etc. as well as non functional quality metrics such as response time, throughput, scalability, etc. The functional testing was conducted by applying various assertions. The Load Testing and Stress Testing was conducted with Simple, Fixed-Rate, Variance, Burst, Thread and Grid strategies to check how Web Services are fulfilling the promises to deliver correct output when number of users are increasing.

REFERENCES

- [1] By Lance Speck, VP and General Manager, Integration Products, Pervasive Software, "Why the data integration is secret to business success", <http://channelmarketerreport.com/2012/12/why-data-integration-is-the-secret-to-business-success/>
- [2] Mr. Manishkumar R. Solanki, Ms. Seema Shah, A Web Service Based Design to Integrate a Web Portal and an ERP System, Int. Journal of Engineering Research and Applications www.ijera.com ISSN : 2248-9622, Vol. 4, Issue 4(Version 1), April 2014, pp.285-288
- [3] Welcome to Apache Axis2 / Java, <http://axis.apache.org/axis2/java/core/>
- [4] <http://itextpdf.com/product/itext>
- [5] <http://www.journaldev.com/2562/java-readwrite-excel-file-using-apache-poi-api>
- [6] <http://www.jfree.org/jfreechart/>
- [7] Charitha Kankanamge, Web Service Testing with SoapUI, Pckt Publishing Ltd., ISBN 978-1-84951-566-5
- [8] Functional Testing, <http://www.soapui.org/Functional-Testing/functional-testing.html>
- [9] Load Test Window, <http://www.soapui.org/Load-Testing/load-test-window.html>
- [10] Web Services penetration testing part 5:Manual Testing with SoapUI,<http://resources.infosecinstitute.com/web-services-penetration-testing-part-5-manual-testing-soapui/>
- [11] Load Testing with SoapUI, <http://learnsoapui.wordpress.com/2011/08/29/load-testing-with-soapui/>
- [12] How to create and execute load tests with SoapUI, <http://university.utest.com/soap-ui-web-service-testing-a-practicum/>
- [13] Kevin J. Ma, Web Services: What's Real and What's Not?, IEEE Computer Society, March-April 2005, pp 14-21
- [14] Jaideep Roy and Anupama Ramanujan," Understanding Web Services", IEEE computer Society, November-December 2001, pp 69-73
- [15] Siew Poh ,Lee Lai Peng Chan, Eng Wah Lee," Web Services Implementation Methodology for SOA Application",IEEE, 2006
- [16] Joe M. Tekli, Ernesto Damiani, Richard Chbeir and Gabriele Gianini, "SOAP Processing Performance and Enhancement", IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 5, NO. 3, JULY-SEPTEMBER 2012, pp 387-403
- [17] J. Leon Zhaoa,Hsing Kenneth Cheng, Web services and process management: a union of convenience or a new area of research?, Decision Support Systems 40 (2005) 1 – 8, www.elsevier.com/locate/dsw
- [18] Enterprise application integration using J2EE, Learn EAI and integrate your applications with Java, By Abraham Kang, JavaWorld.com, 08/09/02
- [19] Enterprise Integration: The Essential Guide to Integration Solutions, By Beth Gold-Bernstein, William Ruh ,Published Jul 16, 2004 by Addison-Wesley Professional. Part of the Addison-Wesley Information Technology Series, ISBN-10: 0-321-22390-X, ISBN-13: 978-0-321-22390-6
- [20] Reiko Heckel, Marc Lohmann," Towards Contract-based Testing of Web Services", Electronic Notes in Theoretical Computer Science 82 No. 6 (2004), <http://www.elsevier.nl/locate/entcs/volume82.html>