Computer Engineering and Applications Vol. 5, No. 2, June 2016

A Proposed Software as a Service (SaaS) Toolkit for Cloud Multi-Tenancy

Mohamed Osman Imam[1], Adil Yousif[1], Mohammed Bakri Bashir[2]

University Science & Technology-Sudan1, Shendi University-Sudan2
adiluofk@gmail.com

ABSTRACT

Cloud computing is a general term for anything that involves delivering hosted services over the Internet. It is a construct that allows you to access applications that actually reside at locations other than your computer or other Internet-connected devices. Software as a Service (SaaS) is a software delivery model in which software resources are accessed remotely by users over the internet. Enterprises prefer using SaaS because of its low cost. SaaS requires sharing of application servers among multiple tenants for low operational costs. Several SaaS Toolkits are available and used to develop cloud SaaS applications. However, most of these Toolkits are very complex. This paper aims to build a new simple and easy SaaS Toolkit to develop SaaS applications and to avoid the difficulties and complexity in the current cloud SaaS Toolkits. The proposed Toolkit is based on java virtual machine and the popular web programming languages HTML and PHP. To evaluate the proposed Toolkit an empirical study has been conducted. The result of the empirical study revealed that the proposed Toolkit outperforms the current Toolkits in terms of complexity, understandability and learnability.

Keywords*:* cloud, SaaS, Toolkit, Athena.

1. INTRODUCTION

Cloud computing emerged as a new computational model to replace the traditional computing model and satisfy the increasing demand for the resources, software and infrastructures[1]. Cloud computing is defined as an on demand service in which shared resources, information, software and other devices are provided according to the client's needs at specific period of time[2].

Developing and maintaining on-premise software consider a complex, costly, and risky task. All software needs hardware, an operating system, a database and Web servers. Once these requirements were provided, a group of developers had to find complex programming Toolkit. Furthermore, a group of network, database, and system management experts are required to keep everything up and running. Unavoidably, a business need would require a change to the software, which would then start a long building, test, and redeployment cycle [1]. Big companies often needed particular facilities to house their data centers. Massive amounts of electricity also were needed to power the servers and the systems to keep them cool. Moreover, a backup site is needed to mirror the data centers to replace them in case of disaster.

NIST defines Platform-as-a-Service as: "The capability provided to the consumer to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, APIs, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment". Nevertheless, as the cloud market grows, PaaS as the deployment model that makes the cloud worthwhile for enterprises and the greatest opportunity to maximize cloud revenue for providers [2]. This paper started by defining the cloud deployment model and service models. The research problem is formulated in section three. The proposed Toolkit is described in section four and its subsections. Section five describes the empirical study and illustrated the results. The last section concluded the paper.

2. CLOUD SERVICE MODELS

Three different types of cloud services are available for cloud clients as described in the following subsections.

2.1 SOFTWARE AS A SERVICE (SAAS)

Software as service provides typically specific, already-developed applications running on a cloud infrastructure. A very famous SaaS is the web-based e-mail. Most software cloud computing services are web-based applications, which can be accessed from different client nodes using a thin client interface, such as a web browser. The clients of software as service cloud do not manage or control the underlying infrastructure and platform; only limited user- settings are available for the customers[3-5].

2.2 PLATFORM AS A SERVICE (PAAS)

Platform as a service cloud offer a managed higher-level application infrastructure, where clients can create and deploy particular types of software and services using the tools, environments, and programming languages supported by the cloud provider. The offers include the use of the underlying infrastructure, such as servers, network, storage, or operating systems, over which the customers have no control, as it is abstracted away below the platform[6-8].

2.3 INFRASTRUCTURE AS A SERVICE (IAAS)

Höfer in[9] described IaaS as "Cloud infrastructure as services in general provide virtualization platforms, which are an evolution of the virtual private server offerings, which are already known for years. The customers buy the resources, instead of having to set up servers, software, and data center space themselves, and get billed based on the resources consumed. They deploy their own software on the virtual machines and control and manage it. The virtual instances can be rented for as long as necessary, which can be as short as an hour" [3, 10].

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

2.4 ATHENA TOOLKIT

Athena Toolkit based on java programming language support multi-tenancy. The Athena Toolkit is an open source application Toolkit for java platform. Since 2008, Jack Li and other developers at Athena source have been developing quite a number of cloud SaaS application for their clients[11]. Initially, they tried many popular Toolkit but gave up all of them due to performance issues and painful configuration [11, 12].

As a mature object relational mapping and persistence Toolkit, Athena provides an elegant solution to object-relational impedance mismatch that enables developers to work with high level objects instead of manipulating SQL statements. Figure (1), "Architecture of Athena Toolkit" illustrates core components in the Athena Toolkit.

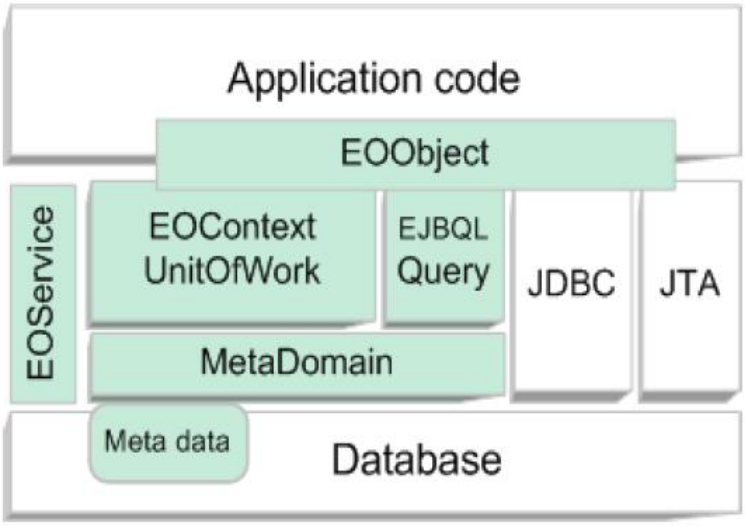

FIGURE (1): Architecture of Athena Toolkit[11]

3. THE PROPOSED TOOLKIT

The Proposed Toolkit is a new Toolkit to develop cloud SaaS applications. The study used object oriented programming concepts and relational database design to build the proposed Toolkit. The Proposed Toolkit is an offline toolkit and software-as-a-service (cloud Software) for rapidly creating business software systems on the cloud. Furthermore, the proposed Toolkit assists in building and deploying core of SaaS applications without the need to write programs from scratch. The proposed Toolkit delivers complete development functionalities with no additional programming needed. However the developer can add some code to customize his applications.

3.1 THE PROPOSED TOOLKIT GUIDELINES

The proposed Toolkit designed to be simpler than the current Toolkits for the software-as-a-service (SaaS). As illustrated previously using the proposed Toolkit is only requiring implementing a few steps to build an application. For creating basic

SaaS application the proposed Toolkit require no programming at all. However, for the developer to customize his application he can use PHP and HTML, a widely used web development tools, to build his applications. HTML and PHP are popular than other web programming languages, they are standardized system for tagging text files to achieve font, color, graphic, and hyperlink effects on World Wide Web pages. The proposed Toolkit employed relational databases because it is standard database model and known by most programmers. Some other Toolkits use object-relational mapping (ORM), however ORM in cloud SaaS Toolkit inherits the limitations and disadvantages of the standard ORM.

3.2 PROPOSED TOOLKIT ARCHITECTURE

The proposed Toolkit architecture as shown in Figure (2) contains five components. The first one is the database component which is based on MySQL database and it is responsible of building, storing and maintaining the SaaS multi-tenancy databases. The second component is the console which is responsible of the operations in the database such as creating new database, connecting the new database with the Toolkit and creating entities in the database. Entities in the Toolkit represent the tables in the database and their attributes represent the table columns. The third component is Interface builder which is responsible for creating and generating the web interfaces for cloud SaaS application. The Fourth component is the web browser which used to display the application. The last component is SaaS application which is the output of the Toolkit.

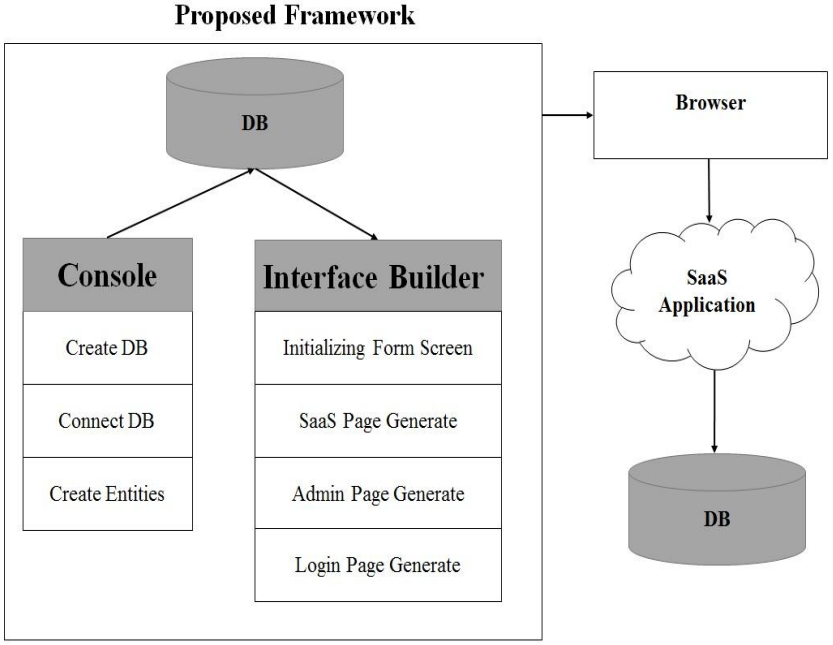

FIGURE (2): The proposed Toolkit Architecture

3.3 PROPOSED TOOLKIT CONSOLE

Console provides utilities for database configuration and database operations as shown in Figure 3. The first step in the console is creating the MySQL database. The next step is connecting the application with the MySQL database to be used in the application. Finally, createing the core tables of the SaaS applications.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

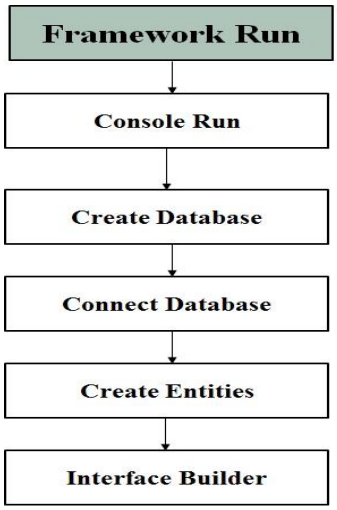

FIGURE (3): The Console of Proposed Toolkit

3.4 PROPOSED TOOLKIT INTERFACE BUILDER

The Interface builder process as shown in Figure (4) started by choosing the fields from tables that have been created in the console. The next step is generating the HTML Form and adding its fields. It generates PHP code and allow the developer to select the operations (insert, select, update, delete) as needed. The last step is generating the admin and login forms.

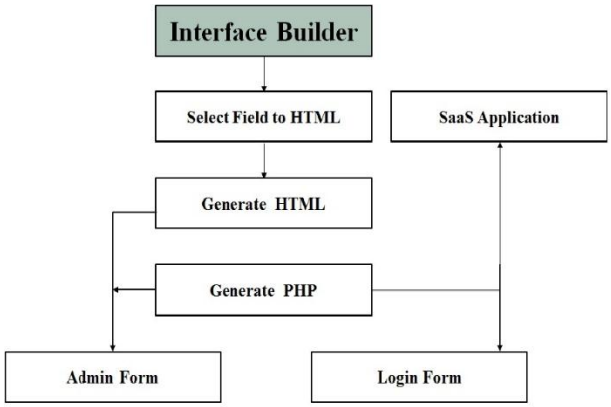

FIGURE (4):Interface Builder of Proposed Toolkit

3.5 Core attributes

Core attributes provides multi-tenancy by adding organization identifier attribute in the table. Making the organization identifier a primary key in the table provides multi-tenancy for the application. If a developer plan to use multi-tenancy, this attribute must be present.

4. EMPIRICAL STUDY

This section explains the results of the empirical study conducted to evaluate the proposed Toolkit. The empirical study selected a sample of 40 students from university of science and technology master of Information Technology. The empirical study has used ISO Usability Standard Model. Besides three important factors that have been added for evaluating the Athena Toolkit and proposed Toolkit.

4.1 USABILITY IN ISO 9126 QUALITY STANDARD MODEL

The ISO/IEC 9126-1 represents the Software Engineering (SE) perspective on usability. In SE, usability is defined as being the capability of the software product to be understood, learned, used and attractive to the user, when used under specified conditions (ISO/IEC FDIS 9126-1, 2000). The Usability sub factors in this standard are understandability, learnability, operability, attractiveness, and usability compliance as shown in Figure (11).

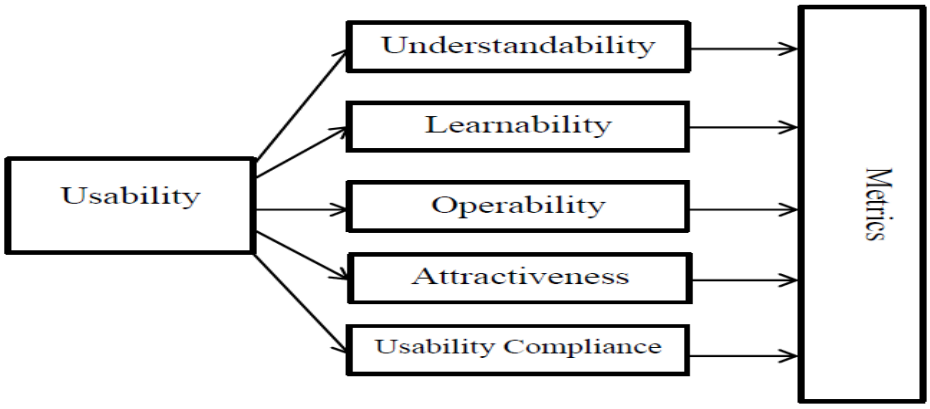

FIGURE (5):Usability in ISO 9126 Quality Standard Model

4.2 ADDITIONAL FACTORS

The additional factors are complexity, exception and error rate and interface creations. These three factors are very important for any development toolkit. As the exception and error rates factor are very important for the evaluation of any programming toolkit. The toolkit to be easy to use it must reduce the error rate and complexity. Furthermore, whenever complexity rate is lower the better usability. The last factor is the interface creation which is also important to build GUI applications.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

4.3 EMPIRICAL STUDY DESCRIPTION

This study has trained students of master of information technology at university of science and technology on Athena Toolkit and the proposed toolkit. The last step of the empirical study was to collect the student's observation about Athena Toolkit and the proposed Toolkit. An analysis of the observation has been carried out to compare the two Toolkits.

4.4 EMPIRICAL STUDY RESULTS

The first observation collected from students was about the rate of complexity.

TABLE 1.
Complexity

Athena Framework			Proposed Framework	
Options	Member Answered	Rate	Member Answered	Rate
Low	5	12.5%	36	90%
Accept	2	5%	2	5%
High	33	82.5%	2	5%

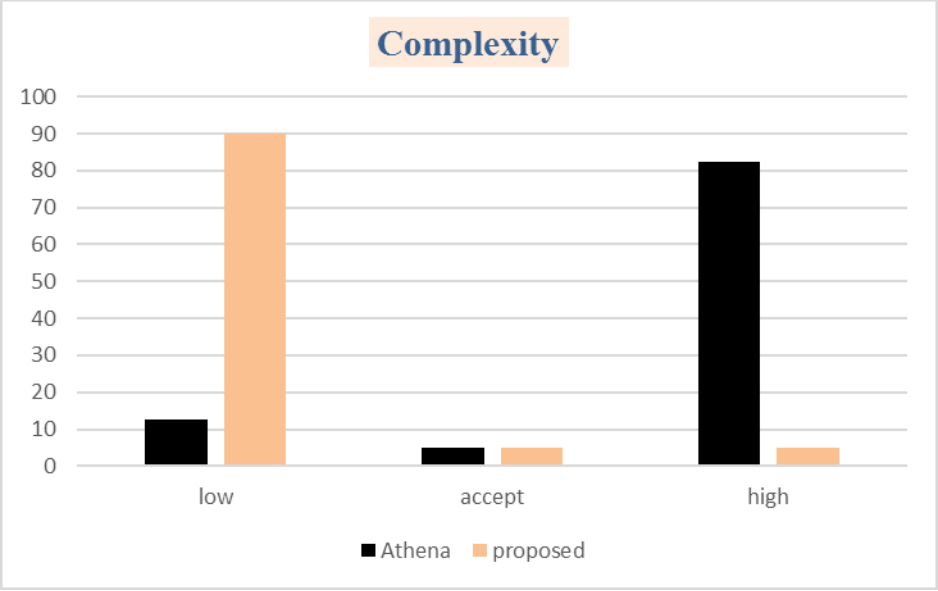

FIGURE 6. Complexity

Table (1) and Figure (6) show that the proposed framework is simpler and has less complexity as 33 percent reported that Athena framework is more complex than the proposed framework. Compared to only two percents reported that the proposed framework is complex.

The second observation collected from students was about the rate of exception and error.

TABLE 2.
Exception and Error Rate

Athena Framework			Proposed Framework	
Options	Member Answered	Rate	Member Answered	Rate
low	11	27.5%	21	52.5%
Accept	6	15%	14	35%
high	23	57.5%	5	12.5%

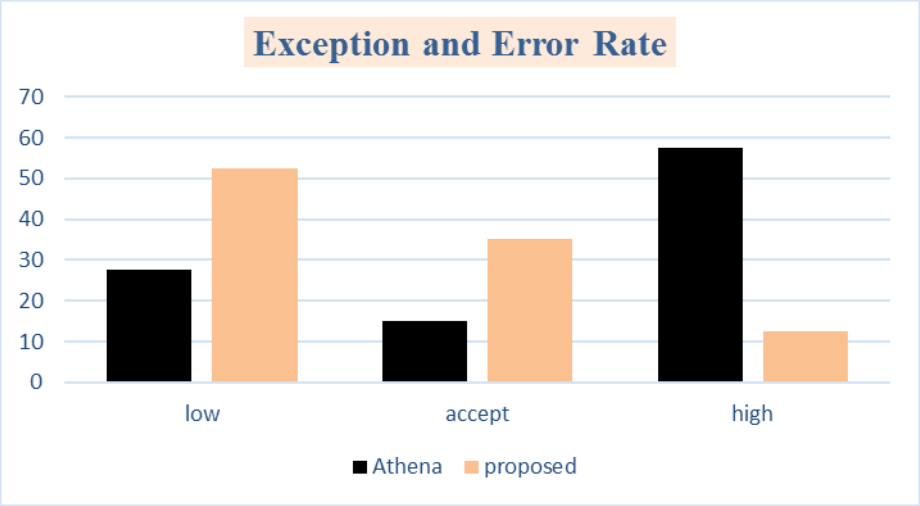

FIGURE 7. Exception and Error Rate

As shown in Table (2) and Figure (7) the result of empirical study reported that, the proposed framework is better than Athena framework in terms of Exception and Error Rate. The majority of IT students say: the exception and error rate in Athena is more than the exception and error rate in proposed framework.

The fourth observation collected from students was about the rate of Learnability.

TABLE 3.
Learnability

Athena Framework			Proposed Framework	
Options	Member Answered	Rate	Member Answered	Rate
Weak	16	40%	8	20%
accept	13	32.5%	3	7.55%
good	11	27.5%	29	72.5%

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

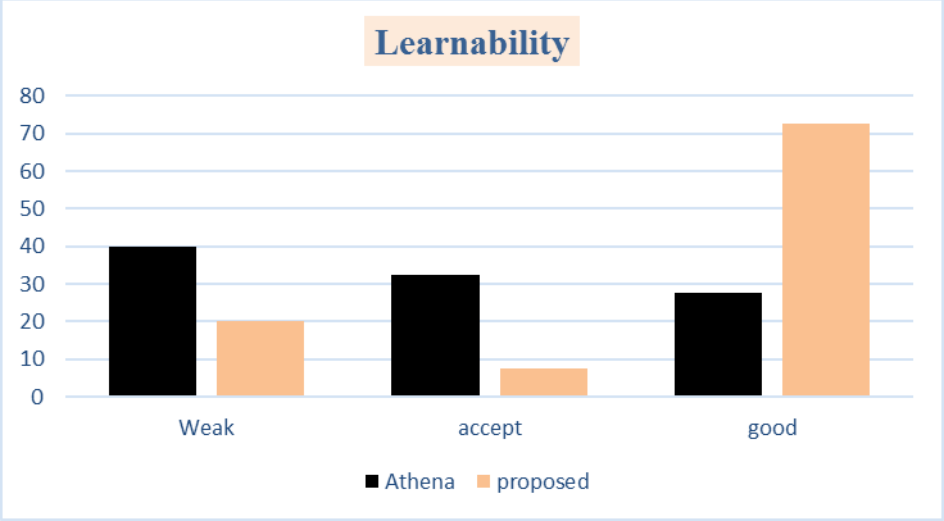

FIGURE 8. Learnability

Regarding Learnability the observation from the empirical study as described in Table (3) and Figure (8) revealed that the proposed framework is easier to learn than Athena framework. As 80% reported that the proposed framework is good in term Learnability process.

The fifth observation collected from students was about the rate of understandability.

Table 4.
Understandability

Athena Framework			Proposed Framework	
Options	Member Answered	Rate	Member Answered	Rate
Weak	15	37.5%	8	20%
accept	12	30%	0	0%
good	13	32.5%	27	80%

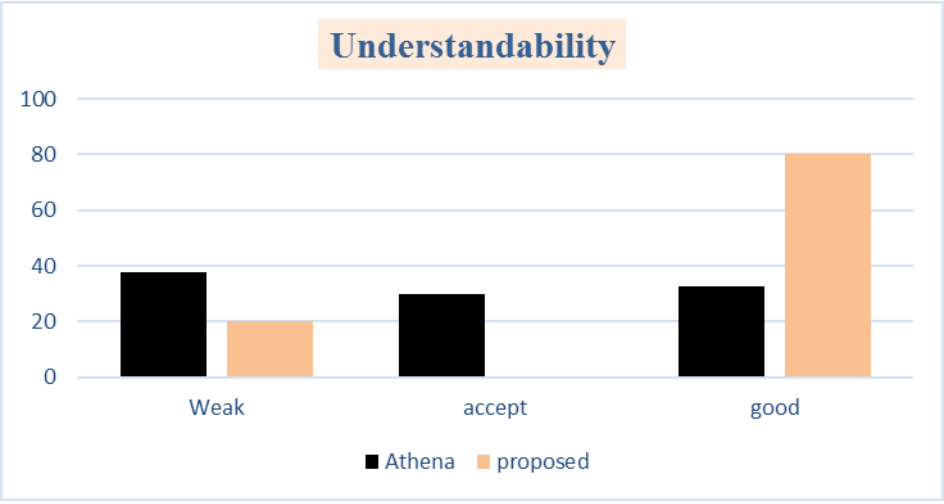

FIGURE 9. Understandability

As shown in Table (4) and Figure (9) the result of empirical study reported that, the proposed framework is more understandable than Athena framework. The majority of IT students say: the understandability in proposed framework is more than understandability in Athena framework.

5. CONCLUSION

This research proposed new software as a service Toolkit to build cloud SaaS applications. The proposed Toolkit has avoided the difficulties and complexity in the current Toolkits. The proposed Toolkit is based on Java, HTML and PHP. The database is based on MySQL database management system. The proposed Toolkit was compared with one of current Toolkits so-called Athena using empirical analysis. The results of the empirical study revealed that the new Toolkit is better than Athena Toolkit in terms of usability, complexity and learnability. The proposed Toolkit is simple for building cloud SaaS software and provides multi-tenancy in the SaaS software in an effective way.

REFERENCES

[1] Amato, A. and S. Venticinque. Multi-objective decision support for brokering of cloud sla. in Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on. 2013. IEEE.

[2] Wu, L., S.K. Garg, and R. Buyya. SLA-based resource allocation for software as a service provider (SaaS) in cloud computing environments. in Cluster, Cloud and Grid Computing (CCGrid), 2011 11th IEEE/ACM International Symposium on. 2011. IEEE.

[3] Bashir, M., et al. An Efficient and Secured Grid-Enabler Interface for Large-Scale Systems. in Cloud Computing (ICCC), 2015 International Conference on. 2015. IEEE.

[4] Paliwal, S., Cloud application services (SaaS)–Multi-Tenant Data Architecture. Infosys technologies limited, [accessed on: 10 Sep 2014], 2012.

[5] Cai, H., N. Wang, and M.J. Zhou. A transparent approach of enabling SaaS multi-tenancy in the cloud. in 2010 6th World Congress on Services. 2010. IEEE.

[6] Abedinia, O. and N. Amjady, Short-term wind power prediction based on Hybrid Neural Network and chaotic shark smell optimization. International Journal of Precision Engineering and Manufacturing-Green Technology, 2015. 2(3): p. 245-254.

[7] Bremer, A.A. and L.A.L. Prado. Municipal m-Services using SMS. in Euro mGov. 2006.

[8] Yousif, A., M. Farouk, and M.B. Bashir. A Cloud Based Framework for Platform as a Service. in Cloud Computing (ICCC), 2015 International Conference on. 2015. IEEE.

[9] Déglise, C., L.S. Suggs, and P. Odermatt, Short message service (SMS) applications for disease prevention in developing countries. Journal of medical Internet research, 2012. 14(1): p. e3.

[10] Erl, T., R. Puttini, and Z. Mahmood, Cloud Computing: Concepts, Technology and Design, 2013, Prentice Hall PTR.

[11] team, A. Open Source Metadata-Based Java ORM Framework for Cloud SaaS Applications. 2011; Available from: http://www.athenasource.org/java/.

[12] Yellamma, P., N. arasimham Challa, and V. Sreenivas, Intelligent Data Security In Cloud Computing. International Journal of Current Engineering and Technology, 2014. 4(1).

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)