SaaS: A New Era for Call Center Based on Cloud Computing

Md. Mahmudul Hasan and Woakil Uddin Ahamed

Abstract—Cloud Computing has become an emerging field of research in recent days for its various cost effective approaches. In this research work, we have developed a live chatting system based on cloud computing to demonstrate its effectiveness in call centers. The developed system follows the core idea of cloud computing, especially the 'Software as a Service' (SaaS) model and its utilization in IT solution companies. The demand of this model is increasing rapidly among the clients because of its cost efficiency and flexibility. Moreover, customer care service providing companies spend a lot of money to buy and maintain chatting software. For this reason, a real time chatting application has been developed in this work in such a way that users of this system can have on-demand and hosted services. The application has been created by using Dot Net framework and written in C# with Asynchronous JavaScript and XML (AJAX) and Language-Integrated Query (LINQ) functionalities. The application has been designed as a web based real time software to those companies, which are worried about expenses and maintenance of the customer care software. The system was simulated in a real time environment to support and solve customers' queries. This cloud based application also guarantees how one can be benefitted using the SaaS model compare to the existing system.

Keywords—AJAX, Application Programming Interface (API), ASP.NET, Cloud Computing, C#, LINQ, .NET Framework, SaaS

---

1 INTRODUCTION

Customer care service is one of the popular aspects in IT sectors due to its excessive demand among the clients to get help or assistance in various sectors such as ISP, e-commerce (e.g. online shopping), bank, IT firms, telecom industries, insurance, health-care (e.g. tel-emedicine), and so on. These companies support their valuable customers' queries through internet or over the phone. In the era of internet and modern technology, most of the companies are trying to boost their sales. Every company is trying to provide a better and easy customer care service for their valuable customers over the phone or via the internet. As a result, internet telephony based call centers are becoming more expensive. Everyone is trying to switch over to a cheaper solution to make the services easier and user friendly with low cost. In this context, customer supports become very expensive and demanding which is gradually affecting the IT sectors. The key to solving this problem is to introduce user friendly and cost effective software, which can support online customers. Nevertheless, the vendor for live chatting software is becoming more costly day by day. In order to support Small & Medium Enterprise (SME) in a competitive market, the application was designed in such a way that the system could solve the problem of excessive expenses for supporting the customers.

The system was created as an economical approach towards the SME IT firms with all facilities and of course with reliability. The project was developed by using .NET framework, which is a well-known platform for security and reliability on web based systems.

1.1 Basics of Cloud Computing

Cloud computing is one of the most prominent technologies to adopt over existing infrastructure because of its plug and play services. It has become a cherry on top of the cream for the companies and researchers for its simplicity and cost efficiency.
In 1990s, at first, Sun Microsystem initiated to promote cloud based architecture by using internet. Moreover, there is no doubt that Application Service Provider (ASP) based services have been popularized by SalesForce.com. And, this company creates a tremendous demand of cloud based solutions with a lot of success stories [1].
There are various types of services which are incorporating with cloud computing such as [2]:

- *MM Hasan is with the Daffodil International University.*
  Email: mhasan@daffodilvarsity.edu.bd

- *Woakil Uddin Ahamed is with the School of Computer Science and Electronic Engineering, University of Essex, UK.*
  Email: wuaham@essex.ac.uk

- Desktop as a service (DaaS),
- Identity management as aservice (IMaaS),
- Communication as a service (CaaS),
- Infrastructure as a service (IaaS),
- Software as a service (SaaS),
- Platform as a service (PaaS) and
- Databases or storage as a service (DBaaS).

Among them, the following three are considered as the core component of cloud computing [3]:

- Infrastructure as a service (IaaS): Infrastructure as a service delivers basic storage such as servers, storage systems, switches, routers and compute capabilities to handle workloads over the network. For example, Amazon EC2, AppNexus, etc.
- Platform as a Service (Paas): It is responsible to develop the environment and storage management. In other words, it provides a compatible Application Programming Interface (API) to look after the implementation. For example, Google App Engine, Windows Azure, etc.
- Software as a Service (SaaS): It provides a complete full function solution through internet. For example, Gmail, Salesforce.com, etc.

1.2 Fundamentals of Software as a Service (SaaS) Model

The acronym SaaS of Software as a Service was first used in an article by Software & Information Industry Association (SIIA) in February 2001[4].It is a concept of using software or any particular applications of a software through internet. The core concept behind this model is that the users of the cloud application do not need to install the software in their premises. In the SaaS model, subscribers are needed to access their desired application from the cloud provider through internet. In general, the users of SaaS application pay charges based on CPU hour.

The remainder of this paper is organized as follows. Section 2 reviews previous researches and current context of customer care. In section 3, the basic of live chatting software and its incorporation with SaaS model are presented. Section 4 describes design, development and implementation of the developed software. Section 5 provides the simulated results and analysis for illustration, followed by concluding remarks in section 6.

2 LITERATURE REVIEW

2.1 Previous Researches and Overview

Software as a Service (SaaS), has received a lot of awareness in recent years when the cloud computing spreads its eminence in the world of IT industry. It has become an emerging trend to all service oriented companies. Although many new technologies were launched to maintain service oriented architecture, SaaS model turn into more popular among them because of its flexibility and adapt-

ability in the existing systems [5].

Now the questions arise how the concept was brought in and why it was also connected to this project?

To answer these questions, it is required to describe from the beginning.

There was no public internet linkage between vendors and customers about 30 years ago. That is why; everything was installed based on premises of the users. In this period, software was manually delivered from the vendors to the end users (e.g. typical format were .exe, .dll, .dmg, etc) to utilize the facilities. The clients were responsible to handle all sorts of risks including buying infrastructures such as operating system, storage, etc.

Internet became available to public about 20 years ago and a new era began with web applications. However, the software was still installed on the customers' end. About 10 years ago, a new concept launched about installation of the applications. To be more precise, the SaaS idea came to the end users with a lot of expectations over existing systems. Everyone accepts this model due to its elasticity and heterogeneity in the existing system [6].

Nowadays, most of the companies are trying to increase their sales by giving a better customer service. By using SaaS model, these companies can save a lot of money instead of using traditional on-premises enterprise solutions.

The most common SaaS applications in the market are [7]:

- Customer Resource Management (CRM),
- IT Service Management and
- Content Management.

The focusing point in this project was to create a Customer Resource Management (CRM) solution, which is broadly used software to communicate with customers. It is an easy and effective enterprise solution for caring customers' queries. The most common CRM solution is customer support based applications. That is why; 24/7 enterprise solution such as a chatting system was preferred in this research work to demonstrate SaaS capability over existing system for call centers.

To do this research, it was obligatory to know the current context of chatting systems in the software market. There are various types of chatting systems available in IT industry. Typical characteristics in these existing systems are sending messages, auto refreshes, save and print conversations and leave the application. For example, LiveChat.com, a Poland based software company, is one of the leading chatting solutions provider in this industry. They support their customers by providing tools and HTML codes on the clients' web page [8] [9]. Other IT giants like Yahoo, Skype, MSN, Meebo, Go to Meeting, Go to Webinar, etc., also provide chatting applications for

instant messaging. The major differences are that in these platforms, users need to install the same application on their desktop (e.g. .exe file, .dll file) or go through from other web sites such asMeebo, eBuddy.com, Nimbuzz, and so on. On the other hand, Chat4support.com and Chatstat.com are dedicated companies to provide the chatting applications between the users over internet [10]. However, some of them need special specifications such as web 2.0 or JavaScript (JSP) enabling on the browsers. ActivaLiveInc., a Michigan based Software Company, is trying to provide some extra facilities to their software such as *Velocity messaging*, which is an automatic quick response technology to the customers' questions through the agent. They are also trying to evolve the technology to identify what has been typed from the users' end [11]. In addition, Facebook, a leading social networking website, uses web based chatting system for chatting and it archives all the messages between senders and receivers [6][15]. Figure 1 shows typical web based chatting solution from LivePerson Inc. to get a glimpse of recent web based chatting application.
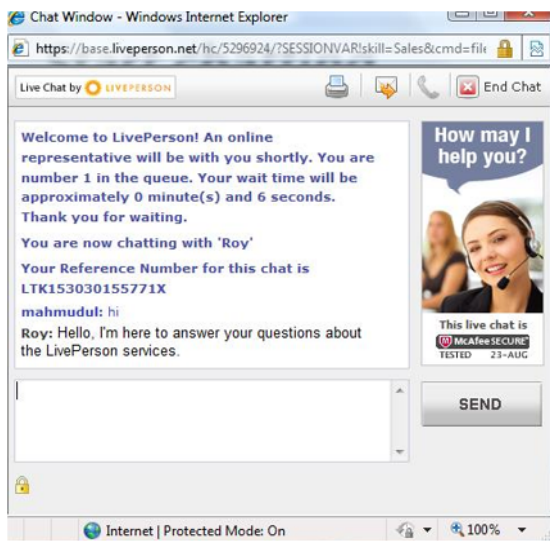

Figure 1: Screenshots of original chat window from Liveperson.net

2.2 Problems in Existing Service Based IT Firms

What are the current conditions to support the customers in service based IT firms?

Why IT Solutions Company need to deploy towards a new technology?

To answer these two questions, it is compulsory to clarify the current situation of supporting customers.

There have been several efforts taken to provide a good solution for customers support in a cost effective way. The following list shows the procedures that are followed by various IT firms to support their customers.

A list of existing customer support systems in the market are [12]:

a) Phone, b) E-mail, c) Online text chatting, d) Call back system and e) Internet telephony.

The most common approach towards customers support is call centre. There is no doubt that this approach is the best in sales and marketing for customer care. However, with this solution, most of the companies are struggling with overhead of driving call centers such as highly equipped infrastructures, trained operators, maintenance cost, etc. Therefore, it becomes more expensive day by day.

For this reason, online chatting or to be more precise, text chatting becomes more popular, because of its easy handling, flexibility and ubiquity characteristics. However, these companies are now becoming more demanding than previous. They provide chatting service based on operators and customized functionalities.

According to their services [8], it is clear that, they charge different prices of their different features such as pay per users or monthly basis subscription fees per operators. As a consequence, it is a bit annoying for both clients and vendors to maintain the software properly.

The main problems of the customer service oriented companies were discovered as:

a) Lower cost efficiency due to on-premise installation, b) Bespoke designing of the software for every client and c) Lower revenue stream.

According to the project, one of the ways to solve the above discussed problems is switching to the SaaS model.

3   What is Live chatting Software?

Live chatting software is a system which is responsible to communicate through internet in a real time environment. There are two types of live chatting software. They are given below [13]:

a. Desktop-based Software: This type of chatting system is comprised with executable file such as .exe, .dll, etc. Therefore, the users need to install and execute the system on their machines. For example, Yahoo Messenger, phonate telecom, Windows Live Messenger, Skype, Local Phone, etc.

b. Web-based Application: Web-based chatting system is the application which runs on user's browser. For instance, DimDim, Meebo, e-buddy, Barclay's Bank Customer Care web site [14], Facebook [15], etc.

This project was built based on web-based architecture, which was written in C# with AJAX and LINQ functionalities [1].

3.1 Why SaaS model in the new system?

Even though a lot of models exist in cloud computing paradigm such as Infrastructure as a Service (IaaS), Plat-

form as a Service (PaaS), and etc., preference was given to Software as a Service (SaaS) model among them based on the following characteristics [16].

a) Lower Cost, b) Reduce run time and response time, c) Minimize infrastructure risk

4   DESIGN, DEVELOPMENT AND IMPLEMENTATION

4.1 Design

4.1.1 Principles of SaaS Design

Anything in cloud must be in a virtual environment. According to this principle, virtualization was the greatest concern to design the system. At first, virtualization was the crucial point to set up the environment in this project. Needless to say, this is a simulation work of a real environment. However, the core technologies that are used in SaaS development had been imagined to implement in this application.

4.1.2 Real Time Application Design

This project was concerned about a live chatting system as a service. As mentioned earlier, the system should be executed in a real time based environment. For this reason, the following aspects were concerned to design the real time environment.

For the execution of a real time based scenario, the chatting session would be able to synchronize with the CPU clock. For this rationale, a timer was used to synchronize with current time of Control Processing Unit (CPU).

The real time application would be able to accurate not only in its logical correctness but also in the time when it is performed [17].Due to this aspect, different browsers were used at the same time to check whereas the system was working perfectly or not according to the CPU time.

4.2 Development

In this sub section of the document, the abstraction level of the project is described. The system is about virtualization which is the key technology of cloud computing. That is why; the whole system is divided in 3separate gears. They are given below:

SaaS subscriber: This component was created as SaaS subscriber for this project. eProfessionals.org.uk [18] web site was created as a customer of SaaS provider. This site was particularly created for the demonstration of the cloud computing scenario.

SaaS provider: This site was created as SasS provider named SX Live Chat Solution which is responsible for providing the chatting application to its subscribers.
User: Users are defined as they are customers of SaaS subscriber. They are typical people who are intended

to be a member of a social organization named eProfessionals.org.uk site for helping their customers.

The high level architecture is presented in figure 2 where 3-tier architecture of ASP.NET 3.5 has been integrated to the web application [19].
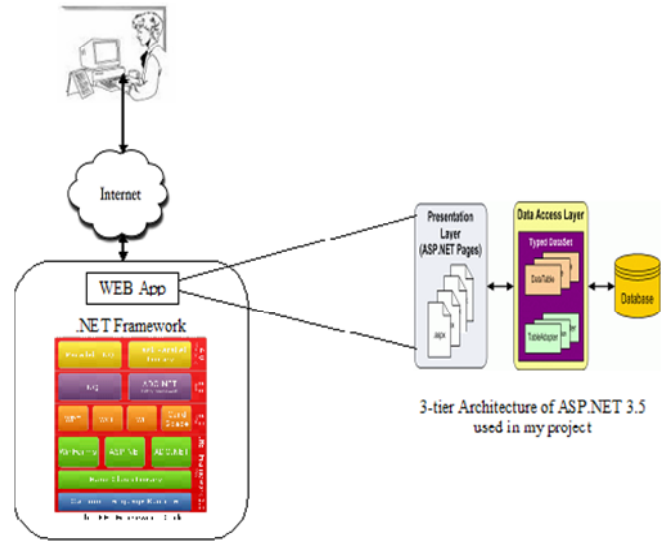

Figure 2: High level architecture of designed sites

4.3 Implementation

To enable the environment of the solution, it is necessary to create the appropriate environment. Here, both of the web sites were created using Active Server Pages (ASP) from Microsoft to dynamically generated web pages [20].

4.3.1 Necessary tools for enabling the environment:

DOT NET Framework [17]: This framework supports Common Language Runtime (CLR) environment which is developed by Microsoft. It is necessary to enable the environment of implanting the system. The following versions of the .Net framework (in Table 1) were used for two web sites:

TABLE 1
Framework Determination of Designed Sites

Name of the Sites	.Net Framework Version
eProfessionals.org.uk	.NET Version 2.0
SX Live Chat Solution	.NET Version 3.5

The rationale for using different domain was to prove the SaaS compatibility in different platforms.

LINQ: Language **IN**tegrated**Q**uery (LINQ) is an object relational mapping system developed by Microsoft. It was used because of it has faster access into databases

which is developed by Visual Studio 2008 with SQL classes.

AJAX: **A**synchronous **J**ava**S**cript and **X**ML (AJAX) is a popular web development technique for interactive applications. AJAX Extensions such as ScriptManager and UpdatePanel were used for timing and maintaining the session of the chatting application. Needless to say, it increased pace and usability of the web pages in the developed system.

4.3.2 Integrating to Web Services

The prime concern was to make a system that could be declared that it follows the principle of Cloud computing specially Software as a Service (SaaS) model.
The following API mechanism for SaaS chatting Application shows the implementation in the developed site:
1. Initial request from user
2. API Call
3. API Response
4. HTML Response

5 Results and Analysis

To check the compatibility and performance of our developed SaaS based chatting application, we have chosen three methods to analyze our outcomes. They are:
a. Integrating the application in a real website to check compatibility,
b. Checking the cost effectiveness and
c. Performance checking in different browsers.

a. Compatibility test:
In this research work, we have integrated our SaaS based chatting application in eProfessionals.org.uk site which is shown in figure 3 to check whether the application is compatible or not to serve multiple clients.

Figure 3: Developed application integrated in a real web site to support its customers.

From the above figure it can be observed that our developed application has been easily integrated to a web site by using the Application Programming Interface (API). And, when a client click the chat button, a window popped up and operator gets a message to initiate the chatting session with the customer. Figure 4 shows a chatting window and it shows ongoing chatting session between an operator and John who is a customer is asking for prices of different packages.
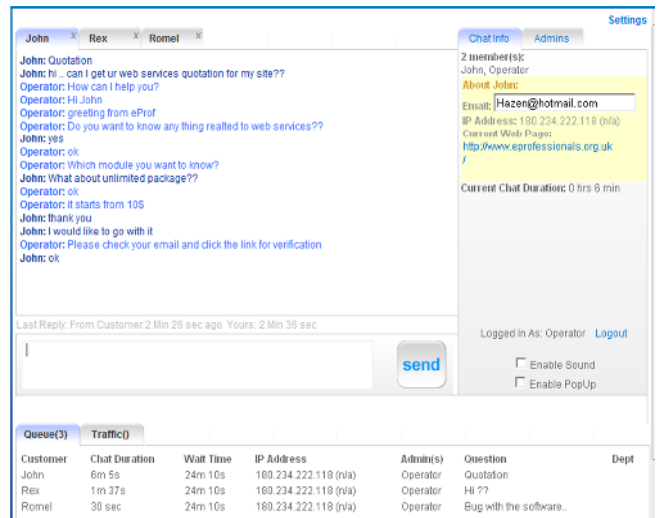

Figure 4: Chatting is going on between operator and customer.

From the above figure, it is clear that this application can be successfully integrated in web sites and it can handle multiple customers simultaneously by putting them in a Queue which is first in first out (FiFo).

b. Cost efficiency test:

Several parameters have been used to determine cost efficiency of the developed system which is shown in table 2.

TABLE 2
Comparison between call center based system and cloud based chatting application

Parameters	Call Center	Chatting Application
To serve Clients	Need one agent for each client	One operator can handle multiple clients simultaneously
Infrastructure	Need individual set up for each agent	One PC is enough to support many clients
Maintenance	Need more support for many agents	Need minimum support since operators are lesser than call center
Payment mode	Needs to support each agent	Rent basis or pay per use

Our pilot deployment in eProfessioanls.org.uk shows that, they reduce 87% expenses after using cloud based solutions rather than call center based system.

c. Performance Test:

For checking the performance of the developed system,

the conversation time between two chatters was required to check. For simplification of the test, different browsers were taken for different users for checking the responsiveness of the system. Several messages and their execution time were taken from Message table to check the performance.

Table 3 shows the conversation and response time between user1 and user 2 using the Internet Explorer and the Google Chrome browsers respectively.

TABLE 3
Response time calculation between two users using IE & Google Chrome browsers

Message ID	Text from User 1	Text from User 2	User 1 Time	User 2 Time	Response Time in seconds(IE & Google Chrome)
1	hi		(S)20/08/2010 05:26:46:00	(R)20/08/2010 05:26:47:06	1.06
2		hello	(R)20/08/2010 05:28:32:57	(S)20/08/2010 05:28:30:06	2.51
3	h r u?		(S)20/08/2010 05:29:21:09	(R)20/08/2010 05:29:22:25	1.16
4		i m fine	(R)20/08/2010 05:30:06:33	(S)20/08/2010 05:30:05:22	1.11
5	hows going ur study?		(S)20/08/2010 05:30:24:43	(R)20/08/2010 05:30:26:23	1.8
6		not bad	(R)20/08/2010 05:30:43:35	(S)20/08/2010 05:30:41:06	2.29
7		what about u?	(R)20/08/2010 05:31:26:25	(S)20/08/2010 05:31:25:09	1.16
8	good?? when u will come to london?		(S)20/08/2010 05:32:03:21	(R)20/08/2010 05:32:04:77	1.56
9		next monday	(R)20/08/2010 05:32:17:41	(S)20/08/2010 05:32:16:01	1.4
10		anyways ,I am not feeling good today	(S)20/08/2010 05:32:55:01	(R)20/08/2010 05:32:56:99	1.98
11		wat happened?	(R)20/08/2010 05:33:26:66	(S)20/08/2010 05:33:25:12	1.54
12	dont know		(S)20/08/2010 05:33:38:12	(R)20/08/2010 05:33:39:88	1.76
13		ok take rest	(R)20/08/2010 05:34:35:48	(S)20/08/2010 05:34:20:00	1.28
14	see u bye		(S)20/08/2010 05:34:27:02	(R)20/08/2010 05:34:28:32	1.3
15		bye,, tc	(R)20/08/2010 05:34:56:56	(S)20/08/2010 05:34:55:04	1.52

Average = 1.562 second
Max=2.51 second
Min= 1.06 second

Figure 5 represents responsiveness between two users according to the data in table 3.
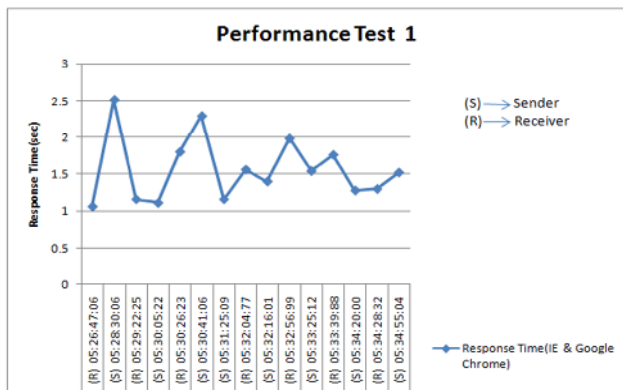

Figure 5: Performance testing of response time of the messages using IE and Google Chrome browsers.

According to the performance test, it was observed that, the response time between the chatters was reasonable. The system did not take infinite time to response. The average time for responding was 1.562 seconds between the Internet Explorer and the Google Chrome browsers.

6   Conclusion

The system is found to be successful in major cases, because the system has the functionality of SaaS application and chatting system. Furthermore, the user interactions of two web sites with different domain such as .Net framework 3.5 and 2.0 have also proved successful for the SaaS based environment and its application in real time software. According to the implementation of the project, it has shown that the SaaS subscribers do not need to install the software on their own premises to support the customers. As a result, it is not necessary for the subscribers to have the related storages for chatting application. For this reason, the system also establishes a cost optimization technique in a service based solution so that different companies could be served with the same code or program which also strictly follows the principles of SaaS model. Nevertheless, the system is not free from its drawbacks. But, it can provide the basic idea of how to implement a SaaS based chatting application in an existing system. To sum up, it can be asserted that the project is successful in its principles.

The future direction of the project is to incorporate it with an intelligent agent for the chatting purposes. Moreover, multi cloud or mobile computing could be used over the SaaS application. However, the system could be designed in a way that it could be free from its weaknesses. In addition, the whole system could be used as a distributed system for more cost optimization.

REFERENCES

[1]   John Viega, McAfee; *"Cloud Computing and the Common Man"*; IEEE CS Digital Library, vol. 42 no. 8, page 106-108.

[2]   Henry E. Schaffer, *"X as a Service, Cloud Computing, and the Need for Good Judgment"*, IEEE CS Digital Library, vol. 11 no. 5, September/October 2009.

[3]   Matt Deacon, *"The Problem with Software- Small is Beautiful, but so is BIG! – In the right place!"* Guest lecturer from Microsoft of CE-811 Module in University of Essex, UK; available at http://blogs.msdn.com/b/matt_deacon/.2010.

[4]   Boucouvalas, A. C., Zhe, Xu, *"Real Time Text-To-Emotion Engine for Expressive Internet Communications"*. International Symposium on CSNDDSP. 2002.

[5]   Turner, M.; Budgen, D.; Brereton, P.; "Turning software into a service", IEEE Computer Society, Volume: 36, Issue: 10, Oct. 2003.

[6]   Michael Miller, *"Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online"*, available at http://www.amazon.com/Cloud-Computing-Web-Based-ApplicationsColab-rate/dp/0789738031/ref=pd_bxgy_b_img_a#reader_078973803.2010.

[7]   John D. McCarthy, M. Angela Sasse, Dimitrios Miras, *"Sharp or smooth?: comparing the effects of quantization vs. frame rate for streamed video"*, Proceedings of the SIGCHI conference on Human factors in computing systems, p.535-542, April 24-29, 2004, Vienna, Austria.

[8]   Mark Handel , James D. Herbsleb, *"What is chat doing in the workplace?"*, Proceedings of the 2002 ACM conference on Computer supported cooperative work, November 16-20, 2002, New Orleans, Louisiana, USA.

[9]   E. Isaacs, A. Walendowski, and D. Ranganathan, "Hubbub: A Sound-Enhanced Mobile Instant Messenger that Supports Awareness and Opportunistic Interactions," Proc. Conf. Computer-Human Interaction (CHI 02), ACM Press, New York, 2002, pp. 179–186.

[10] Budgen, D., Brereton, P., and Turner, M., "Codifying a Service Architectural Style", In Proc. of the 28th Annual International Computer Software and Application Conference, pp. 16-22, 2004.

[11] Bernett, H.; Jaramillo, M.L.; "Assessing Web-enabled call center technologies", IEEE Computer Society, Volume: 3 Issue: 3; May/Jun 2001, Page 24 – 30.

[12] Lassila, A., "Taking a Service-Oriented Perspective on Software Business: How to Move from Product Business to Online Service Business", IADIS International Journal on WWW/Internet, 4(1): 70-82, 2006.

[13] R. Biswas, M.J. Djomehri, R. Hood, H. Jin, C.C. Kiris, and S. Saini, "An Application-Based Performance Characterization of the Columbia Supercluster," Proc. IEEE Conf. Supercomputing (SC), p. 26, 2005.

[14] Oliva, R., and Kallenberg, R., "Managing the Transition from Products to Services", International Journal of Service Industry Management, 14(2):160-172, 2003.

[15] Anthony T.Velte, Toby J. Velte, Robert Elsenpeter, *"Cloud Computing-A practical Approach"*; McGraw Hill Publication; ISBN: 0071626948 / 9780071626941.

[16] D. Gillet, A.V. N. Ngoc, and Y. Rekik, "Collaborative web-based experimentation in flexible engineering education," IEEE Trans. Educ., vol. 48, no. 4, pp. 696–704, 2005.

[17] Herbsleb, J.D., Atkins, D.L., Boyer, D.G., Handel, M., and Finholt, T.A., *"Introducing Instant Messaging and Chat in the Workplace"*. In Proc. CHI 2002, ACM Press (2002), 171 – 178.

[18] Jeffrey Richter, *"Applied Microsoft Windows .NET FRAMEWORK"*; Wintellect Publication; page 5-21.

[19] MM Hasan, Most Tajmary Mahfuz, Woakil Uddin Ahamed, Syed Akhter Hossain, *"Requirement Analysis for Context-Awareness to Create a Digital Persona in eHealth based Pervasive Intelligence Environment"*, Journal of Information and Communication Technologies (ISSN 2047-3168), vol. 2, issue 3, pp.11-15, United Kingdom, 2012.

[20] C. C. Ko, B. M. Chen, J. Chen, Y. Zhuang, and K. C. Tan, *"Development of a web-based laboratory for control experiments on a coupled tank apparatus,"* IEEE Trans. Educ., vol. 44, no. 1, pp. 76–86, 2001.

Md. Mahmudul Hasan is currently working as a lecturer in Daffodil International University, Bangladesh. He has completed his MSc in Computer Science from University of Essex, UK and worked as a research assistant at International Development Academy, UoE, UK. His research interests are in artificial intelligence, cloud computing, intelligent environment and smart applications development.

Woakil Uddin Ahamed is working as a researcher under School of Computer Science and Electronic Engineering (CSEE) in University of Essex, UK. He has completed his BSc in Mathematics from University of Dhaka in 1991 and University of Hull, UK in 2005. He has completed his PhD from University of Hall on "Quantum Neural Networks" in 2009. His research interests include cloud computing and neural networks