Design and Security Analysis of web application based and web services based Patient Management System (PMS)

Sahil Rajput[†] **, Dr S Vadivel**[††] and **Sujala D Shetty**[†††]

Computer Science Department, BITS, Pilani – Dubai, Dubai, UAE

Summary
In this paper, design, implementation, and performance analysis of web application and web services based Patient Management System have been carried out. Further, a comparative study of performances of the above products with different types of securities was conducted. The resultant analysis will be of immense use to the people who design the above types of products.

Key words:
Patient Management System, Java Servlets, Java Server Pages, MySQL, Web Application, Web Services.

1. Introduction

Since World Wide Web is supported across multiple platforms and uses standard communication like Hyper Text Markup Language (HTML) [1], it has become widely popular. Innovations such as Java EE [2] and SSL [3] (Secured Sockets Layer) make it easier to use the World Wide Web as the basis for management systems of dynamic processes. This paper contains design, implementation and analysis of a web based application called Patient Management System. A Patient Management System is used by health care centers and clinics to maintain the records of their patients and patient histories electronically. It provides features like appointment scheduling and test reporting. To increase its usage, a web based approach to the PMS has been discussed in this paper. The design of the Patient Management System was carried out using Object Oriented Analysis and Design (OOAD) approach. The application has been implemented in java language using Java Server Pages (JSP) and Java Servlets using NetBeans IDE [4]. Then, we implement the same using SOAP [5] based Web Services [6]. Since the system contains confidential information, different types of security were incorporated in both the web application and the web service. Analysis includes testing of the web application and the web services for response times. Lastly, the differences in response times when accessing the application with and without security have been discussed. All the data pertaining to the Patient Management System have been stored in a MySQL database. Both the web application and the web services were deployed on local machine using GlassFish Application Server [7]. Till date, not enough research has been done on performance study comparison of web applications and web services, with and without security. Hence, this study was carried out; using a Patient Management System.

Manuscript received November 5, 2008
Manuscript revised November 20, 2008

2. Design of Patient Management System

The design is based on object oriented approach. It includes use case diagram, sequence diagrams, class diagram and finally, the database schema.

2.1 Requirements of PMS

Patient Login:

1. **Patient** *logs in* to **Patient Login System** using PatientID and password.
2. **Patient Login System** checks if the **patient** exists and then, *validates* the **patient**.
3. If **patient** does not exist, **Patient Login System** *creates* new **patient**.

Doctor Login:

1. **Doctor** *logs in* to **Doctor Login System** using DoctorID and password. **Doctor Login System** *validates* the **doctor**.

Schedule Appointment:

1. **Patient** should be able to *check doctor's schedule*, which is available in the **Appointments System**,
2. **Patient** should *schedule an* **appointment.**

Consultation:

1. **Doctor** should be able to *add consultations* with **patients**.
2. **Consultation** should *prescribe* appropriate **tests**. **Test reports** *should be generated* for various **tests** conducted.

Patient History:

1. **Doctors** should be able to *get the personal details* of **patient**.
2. **Patient** should be able *to retrieve* his/her **test reports**.

Texts in bold are potential candidates for classes and texts in italics for methods.

2.2 Use Case Diagram

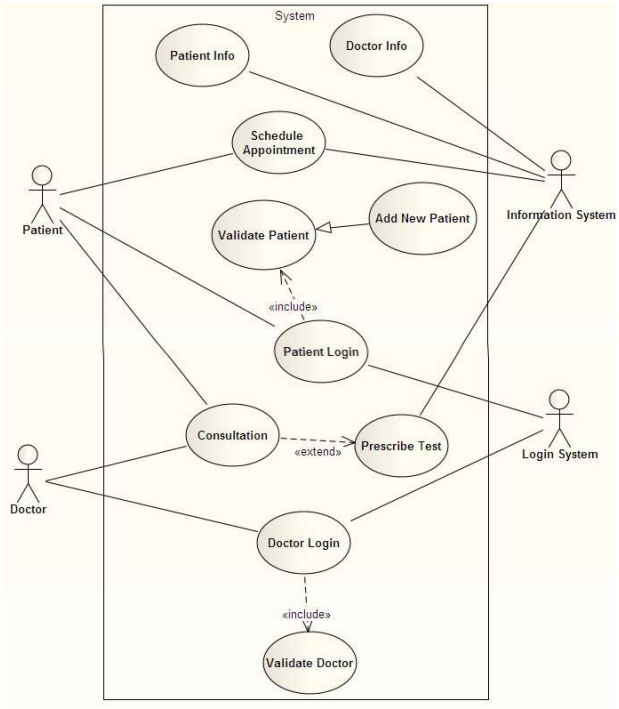

Fig. 1 Use Case Diagram

Doctor, Patient, Login System, Information System and Finance System are the users of the system to fulfill the requirements as discussed above.

2.3 Sequence Diagrams

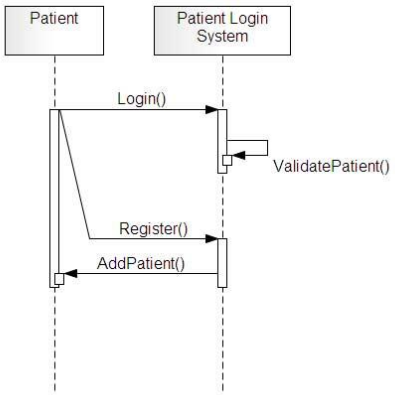

Fig. 2 Patient Login

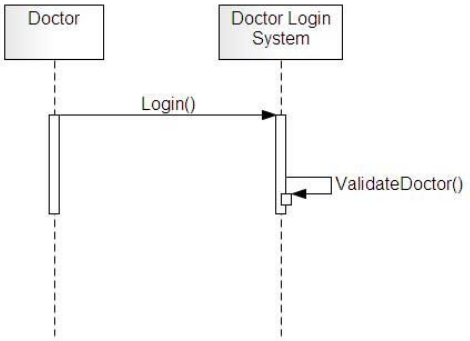

Fig. 3 Doctor Login

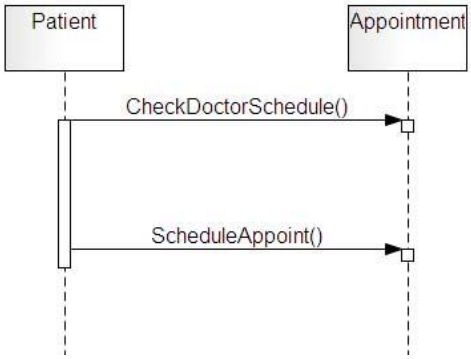

Fig. 4 Schedule Appointment

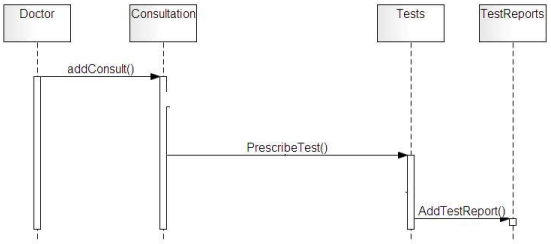

Fig. 5 Consultation

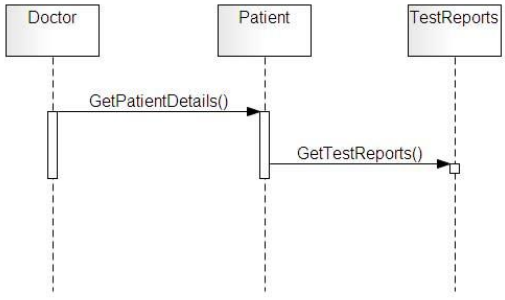

Fig. 6 Patient History

2.4 Class Diagram

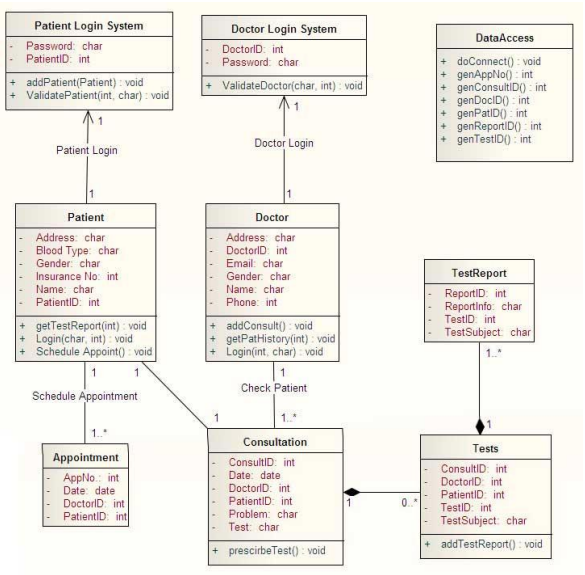

Fig. 7 Class Diagram

2.5 Database Schema

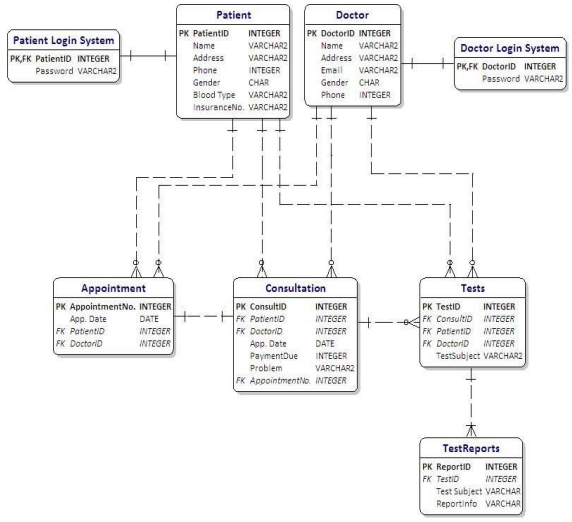

Fig. 8 Database Schema

3. Implementation of Patient Management System

3.1Logging into PMS

When we run the PMS, 3 types of login are available –

- Patient Login
- Doctor Login
- Administrator Login

All the login types require a special ID and password that are stored in the database.

Fig. 9 Logging into PMS

Fig. 10 Patient Login JSP

The above screenshot is the Patient Login JSP. The patient can log in using ID and password or create a new account. At creation of account, patient is given a unique ID.
Similar JSPs have been created for doctor and admin login.

3.2 Patient Activities

Whenever a patient logs in, the image of the patient is displayed that is stored in the database along with the activities that can be carried out by the patient. A patient can do the following:

- Schedule appointment with a doctor
- View current appointments
- Delete an existing appointment
- View prior consultations along with tests conducted
- Log out of the system

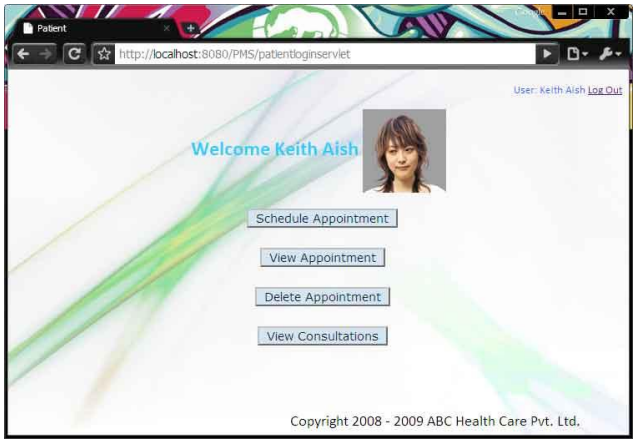

Fig. 11 Patient Activites

Some important features have been included in Schedule Appointment activity. On clicking the Schedule Appointment Tab, a list of doctors appears to select from; and then, a calendar appears to select the date.

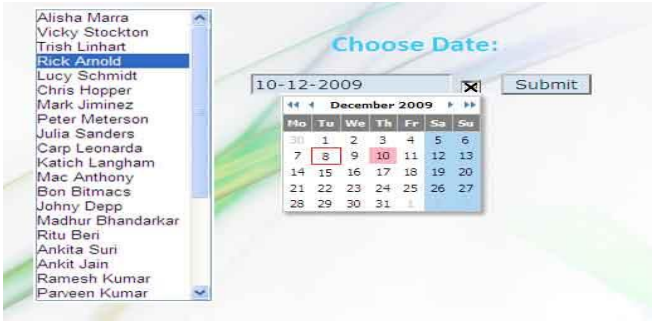

Fig. 12 Scheduling Appointment

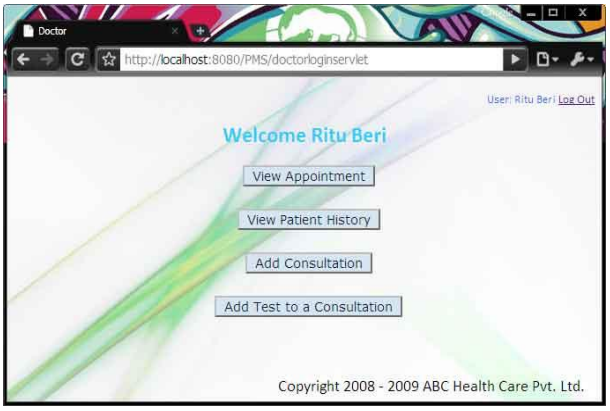

Fig. 13 Doctor Activites

3.3 Doctor Activities

A doctor can perform the following activities:

- View current appointments
- View patient histories
- Add consultation
- Add tests to the existing consultations
- Log out

When viewing the patient history, doctor can search using Patient's ID or name. Then, select the patient.

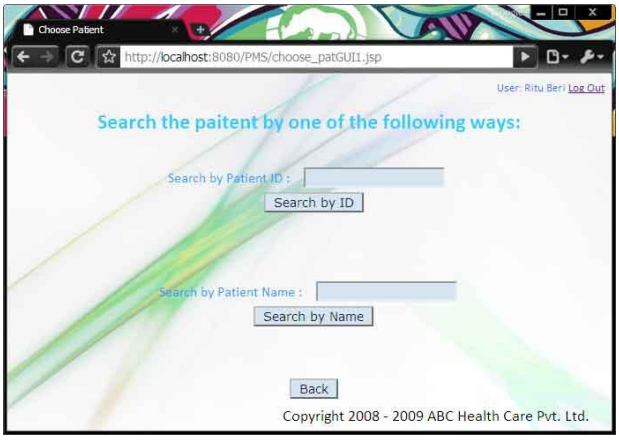

Fig. 14 Patient Search

Fig. 15 Patient History

3.4 Administrator Activities

Admin can perform the following activities:

- Add a doctor to the health care
- Delete an existing doctor
- View all the doctors with their details
- Upload images of patients
- Log out of the system

Fig. 16 Admin Activites

Uploading the image opens a list of patients and then, a dialog box to select the image to be uploaded.

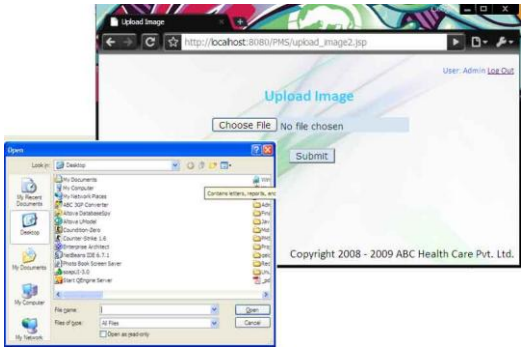

Fig. 17 Upload Patient Image

3.5 Logging out

Session tracking in the system has been done using hidden form fields. To log out, every user has to click the log out link given on the top right of each user session.

Fig. 18 Logging out

3.6 Web Service Implementation

Described above, is the web application based Patient Management System. To develop the web services, each function was made into a web service.
Following is the screenshot of one of the web service clients created. This is a GUI swing client to view the list of doctors.

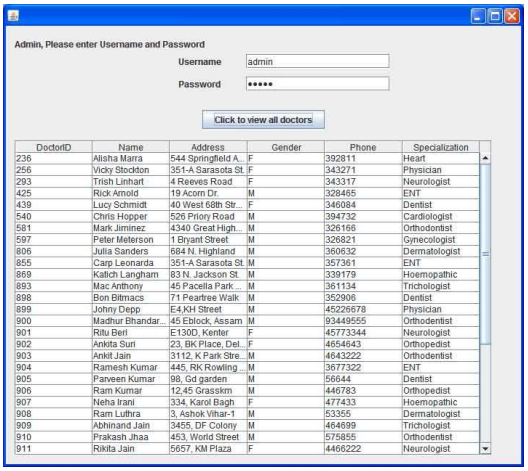

Fig. 19 GUI Client

3.7 Types of Security used

For the web application based Patient Management System, HTTP [8] Authorization over SSL Security was used.

For the web services based Patient Management System, the following security mechanisms were used:

- Username Token with symmetric keys [9]
- HTTP Authentication over SSL
- SAML Holder of Key [10]

3.8 Problems faced in implementation

- In the ViewDoc web service defined above, it can be seen that the return type is a list of objects defined by doctor class. The initialization of this list asks for the number of objects to be created. But, it is impossible to predict the number of objects before creating an SQL connection. Therefore, an approximate number of objects had to be considered.
- It was observed that method ResultSet.getFetchSize did not work and always displayed 0 as the result. Hence, the whole resultset had to be looped to get the number of records.

- Also, when uploading images in JSP pages, the complete path of the image file cannot be forwarded from one JSP to another. Only the name of the image is forwarded. Therefore, a common folder had to be maintained and defined in the servlet.

4. Performance Analysis

4.1 Testing of Web Application based Patient Management System

The web application based patient Management system was tested for response times with and without HTTP Authorization over SSL Security using WAPT (Web Application Testing) [11] tool. Ramp up tests were performed starting with 3 simultaneous clients and going up to 18 clients with steps of 3. The results observed were as follows:

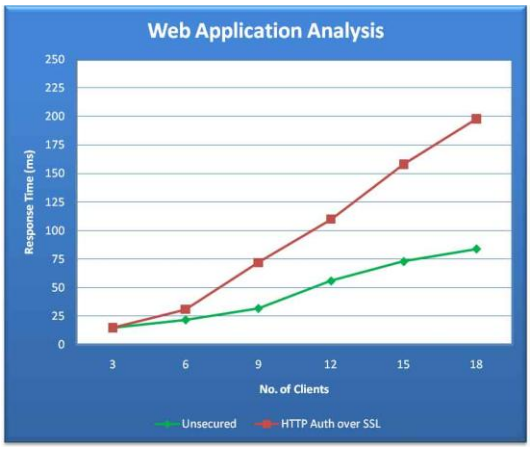

Fig. 20 Web Application Analysis

Average Response Times

- For unsecured web application - 47ms
- For web application with HTTP Auth over SSL – 97ms

4.2 Testing of Web Services based Patient Management System

The web services based patient management system was tested for response times with and without different types of security profiles. The following security profiles were incorporated and tested:

- Username Token with symmetric keys
- HTTP Authentication over SSL
- SAML Holder of Key

The tests were carried out using SoapUI [12] tool. The following results were observed:

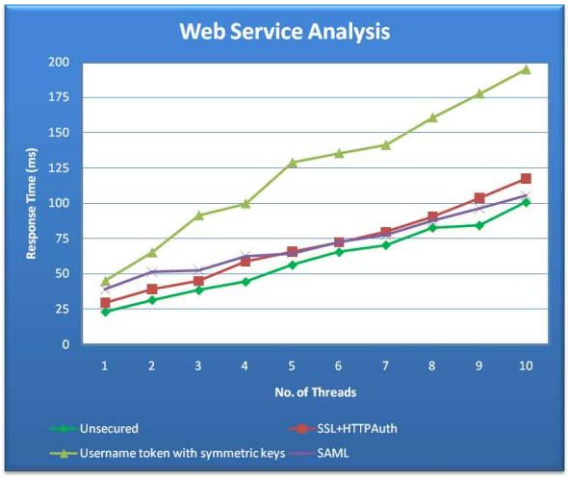

Fig. 21 Web Service Analysis

Average Response Times

- Unsecured web service – 59ms
- SAML Holder of Key – 69ms
- HTTP Authorization over SSL – 71ms
- Username token with symmetric keys – 124ms

4.3 Results

Theoretically, when the number of simultaneous clients increases, traffic at the server increases, which results in overhead, and hence delayed response. From the graphs, it is evident that the response time increases with increase in number of clients. Also, since encryption and decryption processes take considerable time in a secured application (or a web service), the response is further delayed. From the graph, it is clear that the response times for secured applications (and web services) are higher than that of unsecured ones. Hence, the results of the analysis of the web application and the web service are in accordance with the theoretical explanations. When comparing different security profiles, it is found that Username Token takes more time than HTTP Authorization over SSL and SAML Holder of Key. For the Username Token mechanism, the client does not have any key of his own. Instead, it sends its username/password for authentication. A secret key is shared between the client and the server. At runtime, this shared key is generated and encrypted using the service's certificate. However, in SAML mechanism, messages are protected with signed SAML assertions issued by a trusting authority. In SSL, digital certificates are used. Unlike Username Tokens mechanism, clients and servers use their own private keys to encrypt the requests

in SAML and SSL. Since there is extra overhead for generating a new shared key, web service with username token with symmetric keys mechanism takes more time to respond than the web services using SSL and SAML mechanisms.

5. Conclusion

In this paper, design, implementation and security analysis of web application based and web services based Patient Management System have been discussed. The design of the system was carried out using Object Oriented Analysis and Design (OOAD) approach.

The system database contains confidential information like images, phone numbers, addresses and other details. Therefore, to protect such information from intruders, implementation of security is necessary. In the web based approach of the Patient Management System, HTTP Authentication over SSL (Secured Sockets Layer) has been incorporated. For web services based approach, different security profiles have been implemented. These profiles include Username Authentication with symmetric keys, Transport Layer (SSL), and SAML Holder of Key.

For performance testing of web application, WAPT (Web Application Testing) was used. SoapUI tool was used for testing web services by implementing different kinds of security profiles. The results have been compared and presented in the analysis chapter. Results observed in both the web application and the web services were as expected. Firstly, adding any kind of security increased the overhead by a considerable margin. Secondly, it was observed that the overhead (or the response time) increases linearly with increase in traffic (number of clients). It was found that response time of web service implementing Username token with symmetric keys profile increased more steeply with increase in number of threads than the web services with SAML and SSL security.

The results found in this project would be immensely useful when designing web applications or web services. More specifically, the results would help to determine what kind of security should be implemented depending upon the scalability of the web services.

References

[1] Dave Raggett, Arnaud Le Hors, Ian Jacobs, “HTML 4.01 Specification”, W3C Recommendation, 24 December 1999.
[2] Java Enterprise Edition http://java.sun.com/javaee/
[3] Alan O. Freier, Philip Karlton, Paul C. Kocher, “The SSL Protocol Version 3.0”, Netscape Communications, Internet Draft http://www.mozilla.org/projects/security/pki/nss/ssl/draft302.txt
[4] NetBeans IDE http://netbeans.org/index.html
[5] Don Box, David Ehnebuske, Gopal Kakivaya, Andrew Layman, Noah Mendelsohn, Henrik Nielsen, Satish Thatte, Dave Winer, "Simple Object Access Protocol (SOAP) 1.1", W3C Note, 8 May 2000.
[6] David Booth, Hugo Haas, Francis McCabe, Eric Newcomer, Michael Champion, Chris Ferris, David Orchard, “Web Services Architecture”, W3C Note, 11 February 2004.
[7] GlassFish Application Server https://glassfish.dev.java.net/
[8] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, “Hyper Text Transfer Protocol – HTTP/1.1”, RFC 2616, IETF.
[9] Anthony Nadalin, Phil Griffin, Chris Kaler, Phillip Hallam-Baker, Ronald Monzillo, “Web Services Security UsernameToken Profile 1.0”, OASIS STANDARD, 17 February 2004.
[10] Phillip Hallam-Baker, Chris Kaler, Ronald Monzillo, Anthony Nadalin, “Web Services Security SAML Token Profile”, OASIS STANDARD, 1 December 2004.
[11] WAPT http://www.loadtestingtool.com/
[12] SoapUI http://www.soapui.org/

Sahil Rajput is a final year student of B.E. (Hons) Computer Science at BITS, Pilani – Dubai, UAE. His areas of interest are web services and software design.

Dr.S.Vadivel received the PhD degree in Computer Science and Engg from I.I.T Madras, India by 1989.After that he worked in Crompton Greaves in Bombay as research executive for 3 years. Then he worked as Assistant Professor in engg in Govt college at Tamil Nadu, India for 4 years. Then he joined as Research Lead in Think business networks a multinational software company in Tamil Nadu. He has joined BITS, Pilani-Dubai as faculty in CSE by Jan 2003 and currently working as associate professor in CSE in the same institute .He has 10 publications in various international journal and conferences.His current research interest are in web services and security,Embedded controllers ,data mining, and Architecture of enterprise software applications.

Sujala Shetty received the MTech degree from MIT, Manipal in 2002. She is currently pursuing her PhD from BITS, Pilani. She has worked as lecturer in the Computer Science Dept of MIT, Manipal from 1997 to 2002. She is currently working as Senior Lecturer in the Computer Science Dept of BITS, Pilani-Dubai from 2002. She has three publications in International conferences. Her current areas of interest are web services and security.