Journal of Applied Sciences

ISSN 1812-5654

Journal of Applied Sciences 9 (8): 1432-1440, 2009
ISSN 1812-5654

WebML and .NET Architecture for Developing Students Appointment Management System

M.H.N.M. Nasir, S.H. Hamid and H. Hassan
Faculty of Computer Science and Information Technology, University of Malaya,
50603, Kuala Lumpur, Malaysia

Abstract: This study presents the application of Web Modeling Language (WebML) in a Student Appointment Management System (SAMS) to help students and lecturers arrange meetings in an effective and efficient way in a university or college environment. WebML is well designed for web applications and .NET four-tier architecture offers maximum functionality and flexibility in a heterogeneous, web based environment. Each WebML elements is transformed accordingly to Hypertext model as known as site views of SAMS is believed to work best in implementing an appointment management system and turning the traditional management approach into current web technology. Comparisons with existing similar systems are presented based on three typical quality attribute requirements for web applications: usability, performance and maintainability. The usability of SAMS is measured by applying the Goal-Question-Metrics (GQM) approach in questionnaires to collect users' opinions on the system, especially the interface, while the maintainability of this system is measured using the cyclomatic complexity technique. From the survey it is proven that SAMS fulfils the usability requirements as the results show that 67% of lecturers and 75% of students were satisfied with the overall system.

Key words: Appointment management, four-tier architecture, cyclomatic complexity, usability, goal-question-metrics

INTRODUCTION

Recently, with advanced network technologies, mobile phones and cellular routers, it has been possible to access the Internet virtually anywhere. This has brought about an evolution in ways of teaching and registration/administration activities and processes, such as the use of e-learning, e-libraries, online portals for education and so forth. However, this has not occurred in the case of appointment management handling (between lecturers and college students).

There are two common approaches practiced in universities and colleges. Lecturers either put sign-up sheets on their doors (manually) or specify a fixed time period for students to reserve a meeting. These methods do not reflect dynamic changes or cancellations of appointments. There is desire for a Web based appointment management system to solve the above problems.

Currently the development of a web application is a somewhat chaotic and often ad-hoc process lacking systematic techniques and methodologies. As a result, many companies today are experiencing severe problems in the management and maintenance of web applications. Ginige and Murugesan (2001) used the term web crisis to describe this situation. A web based appointment management system utilises web technology to manage the appointments online. Although, there is some web based support in the form of static web pages showing times that lecturers are available for consultation, this does not provide appointment management or any interaction with students. Therefore, it is necessary to refer to similar systems for other services in order to define the features.

Most of the existing appointment management systems reviewed were developed using an ad-hoc approach with consideration of functionalities only. There is little emphasis on the design and development techniques and processes. Only a few appointment management systems adopt UML, which is a widely recognised modeling standard for designing systems. However, due to its complexity and the fact that it has been adapted to a web environment from another sector (software engineering), UML cannot perfectly achieve the goals of Web application development. Web Modeling Language (WebML), which is specified for web applications, can reduce development efforts (cost and time) and allows a more structured development process.

Corresponding Author: Mohd Hairul Nizam Md. Nasir, Faculty of Computer Science and Information Technology, University of Malaya, 50603, Kuala Lumpur, Malaysia Tel: +603-79676435, +603-21784965

Three-tier architecture has been the prevailing architecture for web based systems during the past few years. However, as modern Web applications are becoming more sophisticated, the three-tier approach falls short in several key areas such as lack of flexibility and scalability, inefficiencies and complexities in the development cycle (Behravanfar, 2001). They require advanced features like multi-device access, one-to-one personalization and evolution management, as discussed by Ceri *et al*. (2002). According to Furht *et al*. (2000), by taking an evolutionary step, four-tier architecture, which separates presentation logic from business logic, offers maximum functionality and flexibility in a heterogeneous Web based environment. The presentation server solves the problems brought by the proliferation of devices and content types (Schewe and Thalheim, 2005).

The significance of this project is to turn the traditional approach into a Web based system called SAMS by applying Web Modeling Language (WebML) and .NET four-tier architecture so that students and lecturers are enabled to make better use of web infrastructure and connectivity in order to aim effective automated appointment. This project not only aims to fulfil these functional requirements, but is also designed to satisfy non-functional requirements, namely usability and maintainability.

In this system lecturers can manage consultations over the web; for example they can set up a recurrent consultation time with booking restrictions and maximum appointment volume by specifying dates, times and durations. Students will be able to make appointments, log in (password authenticated), cancel appointments and so on. The system provides appointment reminders and important notices for both lecturers and students. Cancellations by both lecturers and students will be notified in the important notice section.

MATERIALS AND METHODS

The research study was conducted between June 2007 and January 2008 at the Department of Software Engineering, Faculty of Computer Science and Information Technology, University of Malaya. Prior to system development, data gathering was carried out in order to collect information about systems, requirements and preferences. Through the use of powerful search engines such as Yahoo and Google Scholar, more information about similar web based appointment systems, reviews of technologies, design and development processes and techniques and so on can be found to help in determining the system requirements and the development process. The most frequently visited digital libraries such as ACM Digital Library, IEEE Xplore, ScienceDirect, SpringerLink and ISI Web of Knowledge were visited to obtain relevant journals, conference proceedings and articles. Comparative analysis was also carried out to study other similar systems and applications. Five existing similar systems were reviewed and analysed, namely a web-based appointment booking system for a career service, a subsystem of a web-based home schooling application, AppointmentDiary demo system, CyberMatrix Meeting Manager and Timetrade system.

In system development, Web Modeling Language (WebML) and .NET four-tier architecture have been applied in developing SAMS. The WebML addresses the high-level, platform-independent specification and targets websites that require advanced features like multi-device access, one-to-one personalisation and evolution management. By using a four-tier approach, it can integrate existing databases and applications with new functionality for seamless business-to-business integration.

System evaluation has also been carried out to measure the maintainability and usability of the developed system. Cyclomatic complexity and the coupling degree between objects and classes are discussed because these factors directly or indirectly affect the degree of maintainability of the developed system. A questionnaire was used to measure the usability of SAMS. The wide distribution ensured greater anonymity for respondents, which can lead to more honest responses. The Goal-Question-Metric (GQM) approach has proven to be a particularly effective approach for selecting and implementing metrics (Fenton and Pfleeger, 1998). Through GQM, a mixed type questionnaire (i.e., with open and closed questions) was designed.

ANALYSIS OF EXISTING SYSTEMS

All five existing systems, Appointment Booking System for McMaster University (ABSUM), Home Schooling System (HSS), AppointmentDiary, CyberMatrix and TimeTrade, offer basic features. In Appointment-Diary, CyberMatrix and TimeTrade, for example, there are some additional features such as appointment search, automatic cancellation notification, attendance tracking and so on. Only Timetrade provides advanced features like avoiding booking clashes, individual or group appointments and booking restrictions. None of the systems has the following functions: booking deadlines, controlling the volume of appointments (setting a maximum number of appointments for one session) and automatically calculating appointment duration. All the above five systems allow the users to choose from available time-slots.

No existing web based appointment management system specified for lecturers and college students was found in the review of the literature. Similar systems in the current market cannot fulfil all the users' requirements such as setting booking deadlines, restricting the maximum number of students during the consultation time, automatically calculating the appointment duration and so on. There are three typical quality attribute requirements for web applications: usability, performance and maintainability. Table 1 shows a comparison between existing similar systems based on these criteria.

From the Table 1, it can be clearly seen that the similar systems not only fall short in terms of functional requirements but also cannot fully satisfy the non-functional requirements due to the architecture and techniques used. Three-tier architecture has significant drawbacks. It does not truly separate the application into specialised, functional layers. It has too many functions grouped into business logic, the middle tier. Because of that it falls short in several key areas; for example it has inefficiencies and complexities in the development cycle and a lack of flexibility and scalability.

With respect to modelling language, most of the systems do not use any formal language because it is complex and time consuming. Only one system fully utilises UML for the design process. UML is a widespread language used by industries to model systems in different areas such as software engineering modelling and business process modelling. However, due to its general purpose nature and unnecessary complexity, it is not best suited for web application development.

Table 1: Web quality factor comparison

Web quality factors	Appointment booking system	Home schooling system	Appointment diary system	Cyber matrix	Time trade
Usability					
Visual calendar	√	√	√	√	√
Easy to learn	√	√	√	√	
Ease of navigation	√	√	√		
Provides feedback	√		√	√	
Help function	√			√	√
Performance	Limited	Limited	Poor	Poor	Good
Maintainability	Limited	Limited	Limited	Limited	Good

In response to this need, Ceri *et al*. (2007) mentioned that the W3I3 project (funded by the European Community under the Fourth Framework Program) is focusing on intelligent information infrastructure and has produced a novel web modelling language called WebML. WebML is a design methodology that fully exploits the conceptual modelling approach of software engineering from ideas to application (Ceri *et al*., 2002). It addresses the high-level, platform-independent specification and targets websites that require advanced features such as multi-device access (PCS, PDAs and HSPDA phones), one-to-one personalisation (My Yahoo, MyCDNow) and evolution management.

WebML guarantees the model-driven approach, which can reduce development efforts (cost and time) and allows a more structured development process. WebML, which is customised for Web application design, needs further promotion in industry. WebML models such as data models, hypertext models and user models (Fig. 1) are designed using WebRatio for SAMS. WebRatio is a CASE tool to support the WebML design process. It offers a visual environment for drawing the data and hypertext conceptual schemas. Figure 1 shows a brief preview model of WebML application.

Most of the existing systems use either two-tier or three-tier architecture. Only a few large and formal systems apply four-tier architecture. These systems were chosen due to the greater structure and completeness conferred by using three-tier or four-tier architecture. By introducing a separation of business logic from the client and database tier, a three-tier or four-tier approach

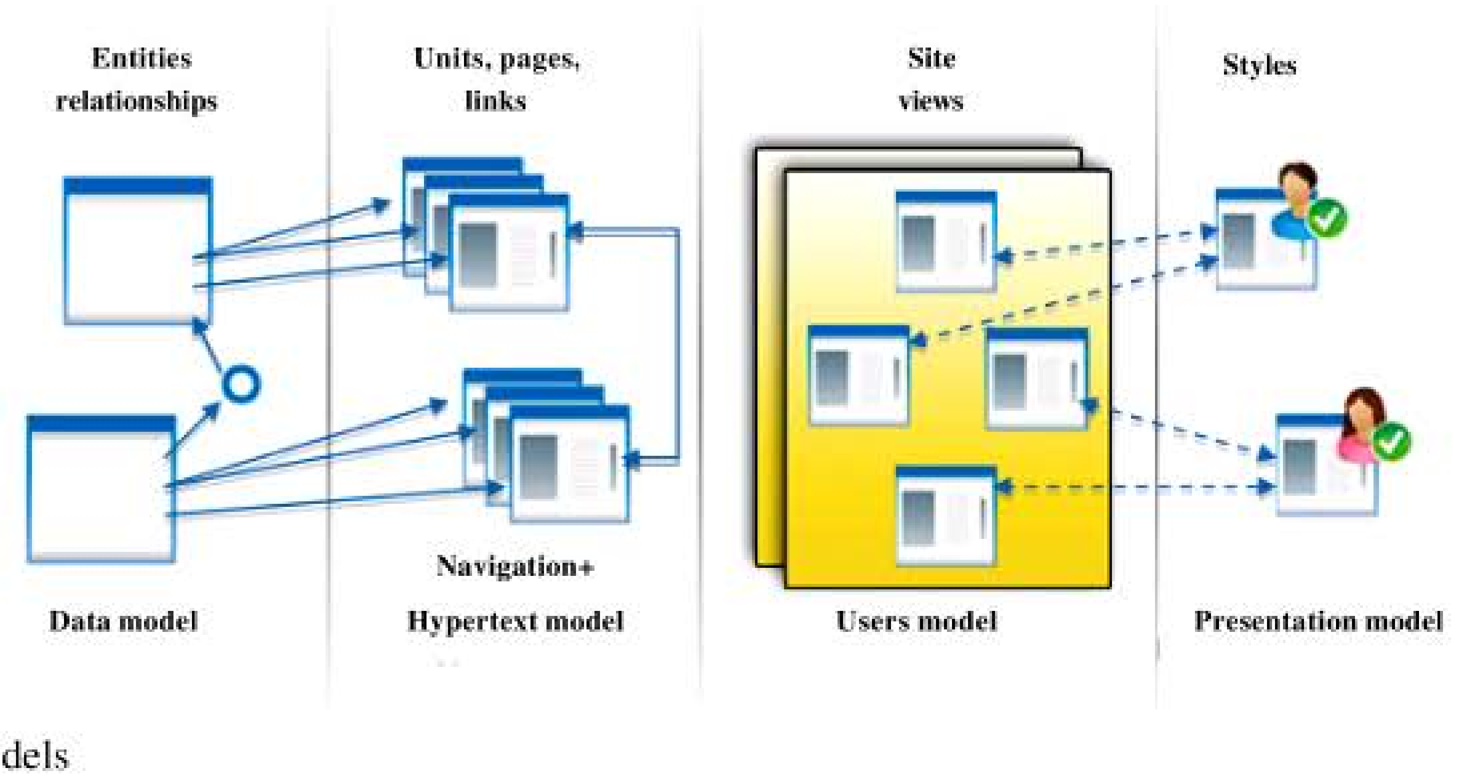

Fig. 1: WebML models

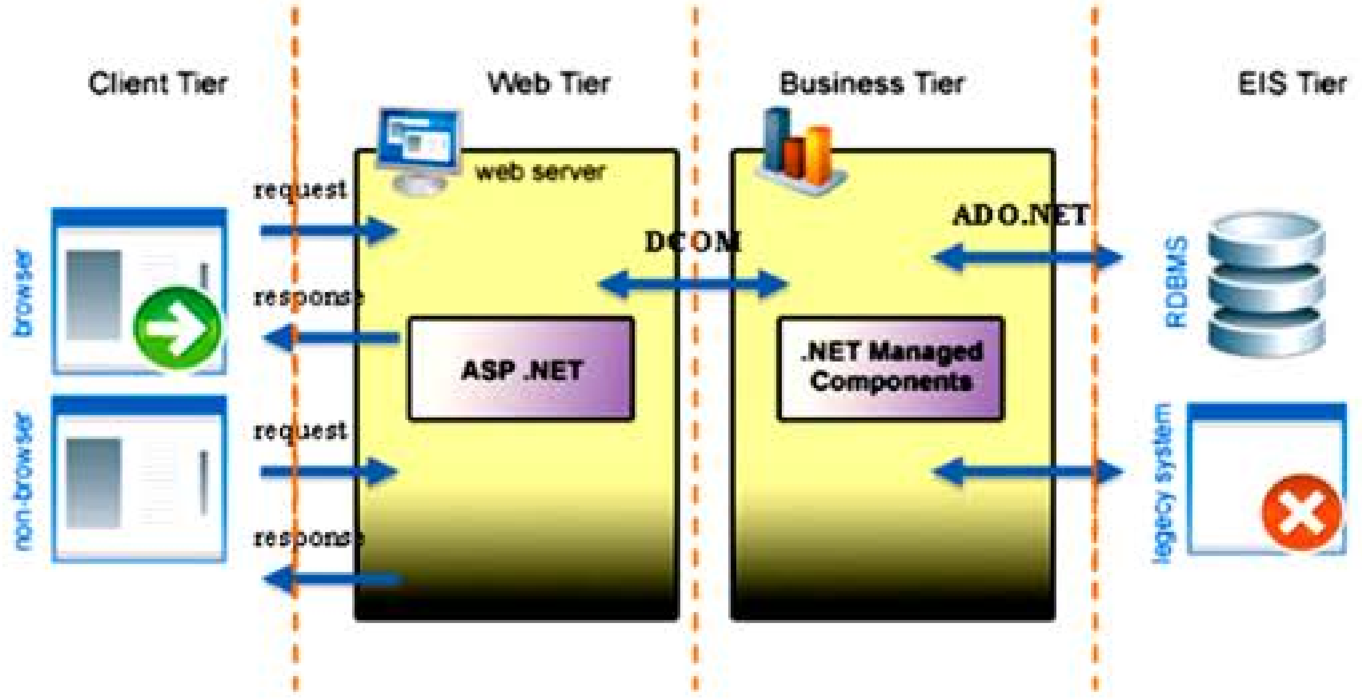

Fig. 2: .NET four-tier architecture

improves flexibility and modifiability compared to two-tier architecture. Applications are easier to manage and change without affecting the other tiers. .NET four-tier architecture comprises the client tier, web tier, business tier and EIS tier (Fig. 2). The four tiers together form the database independent, GUI-based framework that can be used for efficient and rapid application development.

A .NET application can integrate existing databases and applications with new functionality. It provides seamless business-to-business integration. The client tier provides the visual interface for presenting and gathering data. It can be a standard Web browser or non-browser interface. The web tier uses ASP.NET to compile the server-side code in one or a few DLL files on the Web server. It allows for easy and flexible development of Web based interfaces. A number of server components are available for inclusion in the interfaces. The business tier uses. NET Enterprise Services (e.g., COM+ services, web services) to achieve a scalable and flexible architecture. It can interface with the web tier or with other third party applications. The business objects built into the .NET framework provide ease of development and deployment. The Enterprise Information System (EIS) tier is the enterprise information infrastructure. It includes the rational database management systems, active directory services interfaces, transaction processing monitors, legacy enterprise applications and so on.

The four-tier approach further evolves today's three-tier system architecture. By separating the presentation logic from the business logic, it allows for the separation of concerns between the content to be rendered and the rendering logic. The presentation server solves the problems brought by the proliferation of devices and content types. The business logic and data can be centrally exposed using HTTP and XML and thereby it allows any device anywhere on the Internet to access the data and business logic.

In addition, by dividing the work between software and user interface developers, four-tier architecture will cleanly separate development efforts and drastically streamline the development process. It also offers a variety of application services to speed up business transactions and offer additional services. So far, four-tier architecture offers maximum functionality and flexibility in a heterogeneous web based environment.

SITE VIEWS OF SAMS

A site view is a coherent hypertext to serve a well-defined set of requirements. In large applications site views can be hierarchically decomposed into areas, which are clusters of pages with a homogeneous purpose. There are three site views designed for SAMS as shown in Fig. 3a-c.

RESULTS AND DISCUSSION

The methods and processes of web application development are nowadays often done in an ad-hoc style which returns lower quality products as a result. Ultimately, many companies are experiencing problems in managing and maintaining Web applications since the products lack of design and documentation. This unsystematic technique challenges the existing tools and approaches that are ready made for normal desktop software production. In response, this paper suggests the application of Web Modeling Language (WebML) and .NET four-tier architecture in appointment management

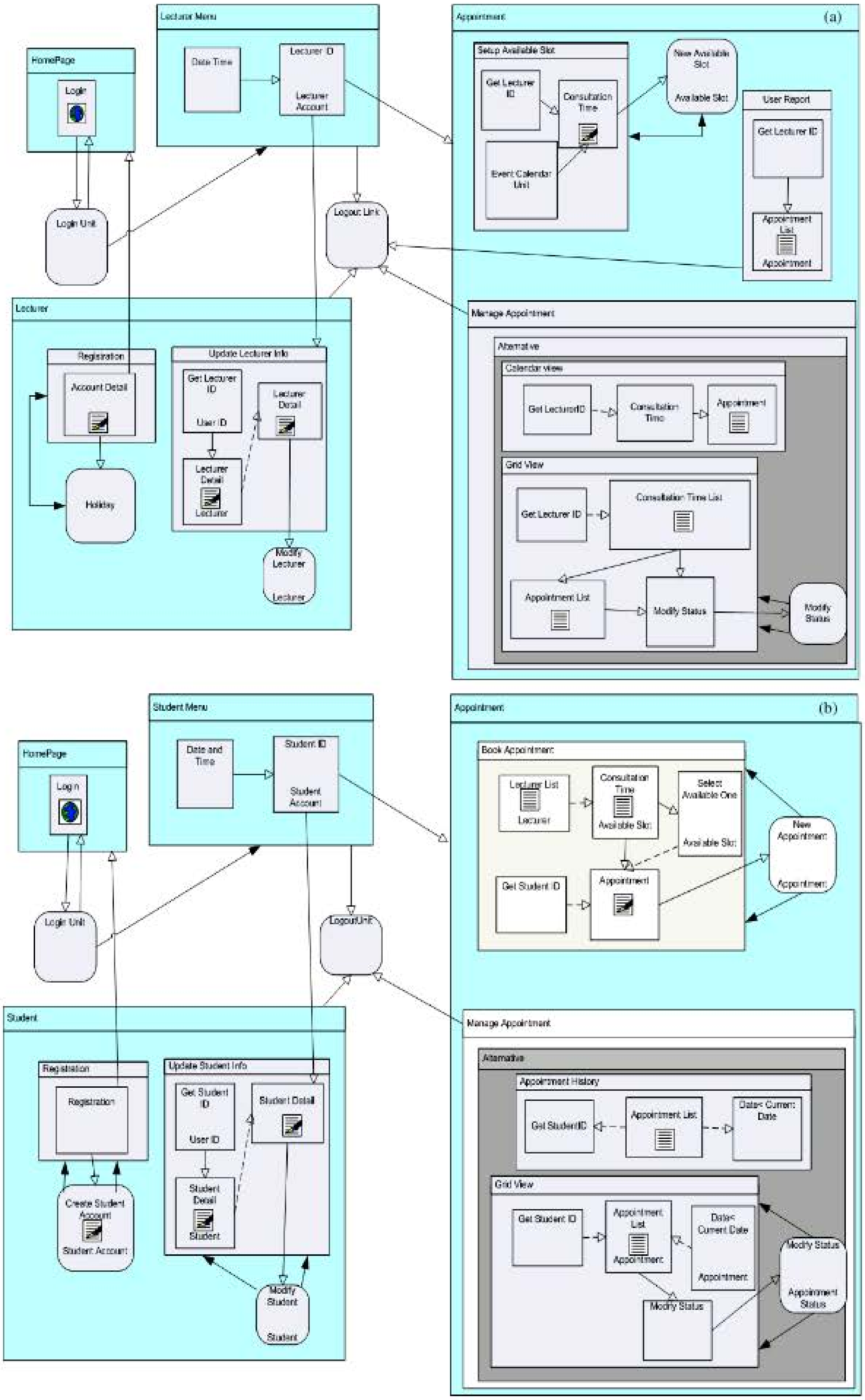

Fig. 3: Continued

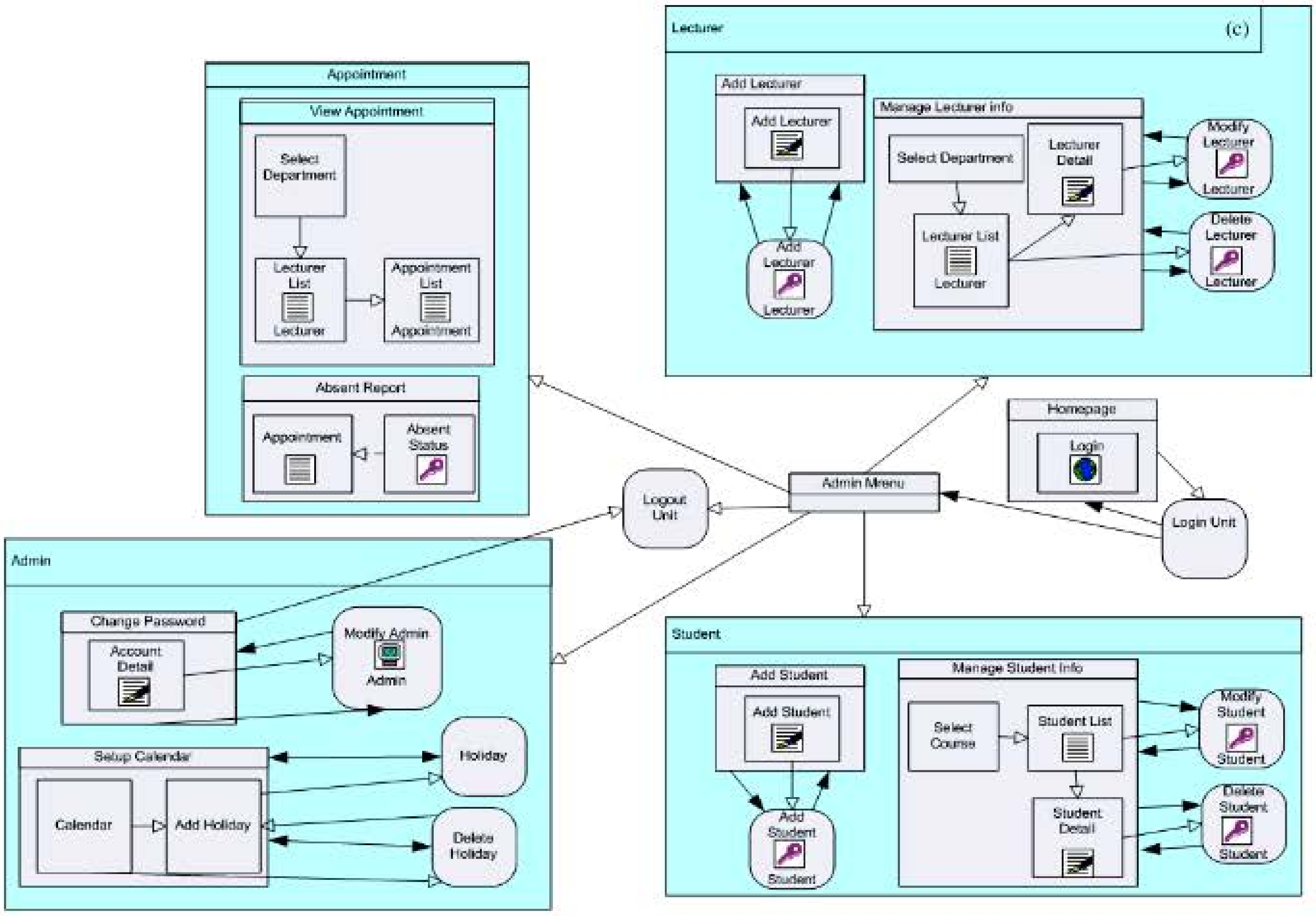

Fig. 3: Three sites view of SAMS, (a) In lecturer site view, there are two areas: Lecturer and appointment in the lecturer site view. Lecturer area consists of two pages: Registration and UpdateUser profile. The particular lecturer's appointment lists can be retrieved from database and displayed as index unit, (b) In the student site view, there are two areas (appointment and student). Registration page and update student information page are included in the student area. The appointment area contains make appointment page and manage appointment page and (c) In the administrator site view, there are four areas: admin, appointment, lecturer and student. Each area comprises two pages. For example, AddLecturer page, through lecturer entry unit, administrator can add lecturer account. If succeed (OK link green color), the information will be stored into the database and display the successful information. Else (KO link red color), the error message will appear in the AddLecturer page

handling in university and college environments to help students and lecturers arrange meetings in an effective way. Here, the results on the maintainability and usability of SAMS are presented.

A typical study shows that on average 60% of human resources in software organisation are spent on software maintenance and 40% are spent on the development of new applications. Therefore maintainability is considered a significant non-functional requirement here. Maintainability testing is carried out to use appropriate metrics to check whether SAMS satisfies that requirement. Software metrics can provide an insight into the software and the processes used to maintain it. The cyclomatic complexity, weighted methods per class using cyclomatic complexity, Coupling Between Object (CBO) class and response for a class are discussed because these factors directly or indirectly affect maintainability, which is one of the non-functional requirements.

The cyclomatic complexity formula is shown below, where, E is the number of edges in the flow graph, N is the number of nodes in the graph and X is the number of exits from the program.

$$\text{Cyclomatic Complexity (CC)} = E-N+2$$

As Cyclomatic Complexity (CC) has enormous impact on the testability and maintainability of code, it was selected to calculate WMC in order to predict the number of tests for a class and whether the class is risky. The number of tests for a specific method is equal to the CC measure of the method. Therefore, if the CC value increases, it is more risky and requires more effort to test.

Table 2: Cyclomatic complexity values analysis 1

SAMS	Median	Mode	Max.	Min.
CC value	2	1	9	1

Table 3: Cyclomatic complexity values analysis 2

CC value	1	2	3	4	5	6	7	8	9
Methods quantity	66	20	30	9	3	5	2	0	1

Table 4: Impact of cyclomatic complexity values

Cyclomatic complexity	Risk complexity
1-10	A simple and low risk program
11-20	More complex, moderate risk
21-50	Complex, high risk
51+	Most complex and highly unstable method

The cyclomatic complexity for the cancellation consultation method in the business tier is calculated as below:

$$CC = 8-7+2 = 3$$

There are a total of 136 methods with coding in SAMS. It is calculated that the maximum CC in SAMS is the add consultation method for which the value is 9. The CC value for most of the methods is 1, as shown from both Table 2, 3.

Various researchers and studies have suggested that a cyclomatic complexity value of 10 or higher for a particular method is considered complex (Card and Agresti, 1988). Table 4 shows the impact of CC values. From the Table 4, it is concluded that a SAMS for which the maximum CC value is less than 10 is low risk.

In Weighted Methods per Class (WMC) using cyclomatic complexity, consider a class C_1 with methods $M_1,. .., M_n$ that are defined in the class. Let $c_1,. .., c_n$ be the complexity of the methods. Then:

$$WMC(C) = \sum_{i=1}^{n} c_i$$

The WMC metric is the sum of the complexities of the class methods. It is a predictor of how much time and effort is required to develop and maintain the class. Because WMC measures both the number of methods and their complexities, it has been proven to be useful in predicting maintenance and testing effort (Kim *et al*., 1995). The larger the number of methods in a class, the more time and effort are required to develop and maintain the class. The WMC using cyclomatic complexity is calculated as:

$$\begin{aligned} WMC = {} & Cc\ (gridview)+CC\ (calendarview)+cc\ (dayrender) \\ & + cc\ (selectchange)+cc\ (show)+cc\ (updating) = \\ & 1+1+3+1+1+3 = 10 \end{aligned}$$

Table 5: WMC analysis

SAMS	Median	Mode	Max.	Min.
WMC	8	5	89	1

Table 6: Coupling Between Objects (CBO) analysis

SAMS	Max.	Min.
CBO	3	0

Table 7: Response for class analysis

SAMS	Max.	Min.
RFC value	65	1

Most classes in SAMS have a WMC of less than 20. There is only one class, the database component, with the highest WMC of 89 (Table 5). According to the study by Rosenberg and Hyatt (1998), a class with a WMC greater than 100 is required for inspection and revision. So SAMS is considered to have passed.

Coupling Between Object (CBO) is a measure that counts the number of other classes to which the class is coupled. It is useful to determine how complex the testing of various parts is. The larger the number of couples, the higher the sensitivity to changes in the design, making maintenance more difficult. In four-tier architecture, layer n may only use the services of layer n–1. Therefore the classes in the presentation layer are only coupled with the business layer, which has just 3 classes. Therefore, the maximum value of CBO is 3 (Table 6).

In SAMS, most of the classes are self-contained. The maximum CBO is 3. Hence, it is concluded that classes in SAMS are relatively easy to maintain.

Response For Class (RFC) captures the size of the response set of a class. It can be calculated as the number of local methods plus the number of methods called by local methods. It is an indirect indicator of maintainability. The larger the number of methods that can be invoked from the class, the greater the complexity of the class and hence more difficult it is to maintain. For example, there are four local methods (lcheckavailable, scheckavailable, sregister, lregister) in the registration class. Each method calls one other method from the user component. The calculation is:

$$RFC = 4+4 = 8$$

In SAMS, RFC for most classes is less than 10. The one outstanding class is the class in the data access layer with the highest RFC value, 65. Similarly to WMC, if the RFC value is bigger than 100, a revision is needed. Here SAMS with a maximum RFC value of 62 is considered to pass. From the results of the tests using this suite of metrics shown in Table 7, it is concluded that SAMS achieves the maintainability requirements.

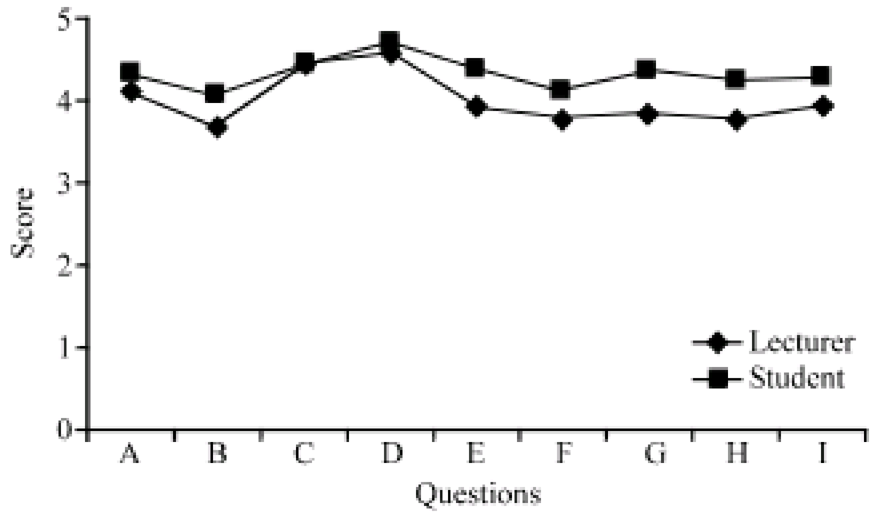

Fig. 4: Bar chart of system usability evaluation

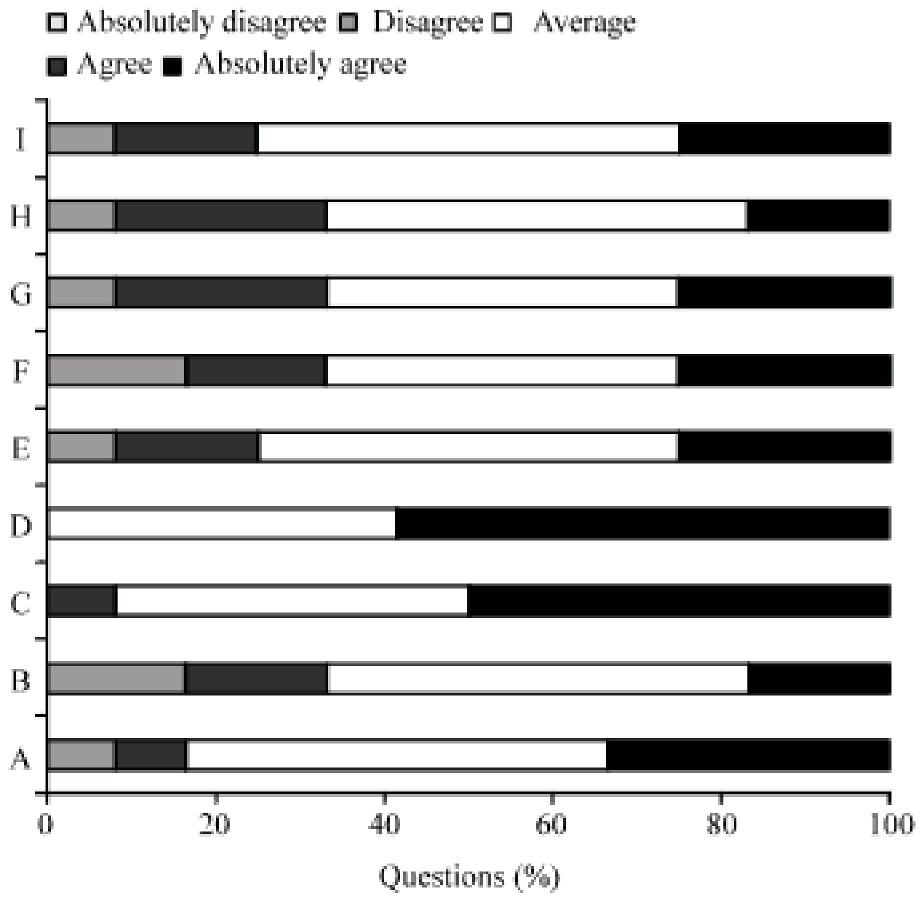

Fig. 5: Bar chart of lecturers' opinions on system usability

An evaluation was conducted to measure the usability and to determine the effectiveness of the interface. A questionnaire applying the Goal-Question-Metrics (GQM) approach was used to collect users' opinions on the system, especially the interface. The lecturers and students in the Department of Computing and Language Centre of KBU International College were selected to test the system. The reason for the choice of these two departments was to get a balanced range for distributing the questionnaire. Students in the language centre are pre-university students with relatively little IT knowledge, while people in the Department of Computing have strong IT backgrounds. The different levels of users involved (in terms of PC knowledge) ensured the dependability and authenticity of this survey. Each chosen user tested the system individually and filled in the questionnaire.

Figure 4 clearly define that the responses from lecturers and students were above the average value for all questions, range from 3.5 to 4.5.

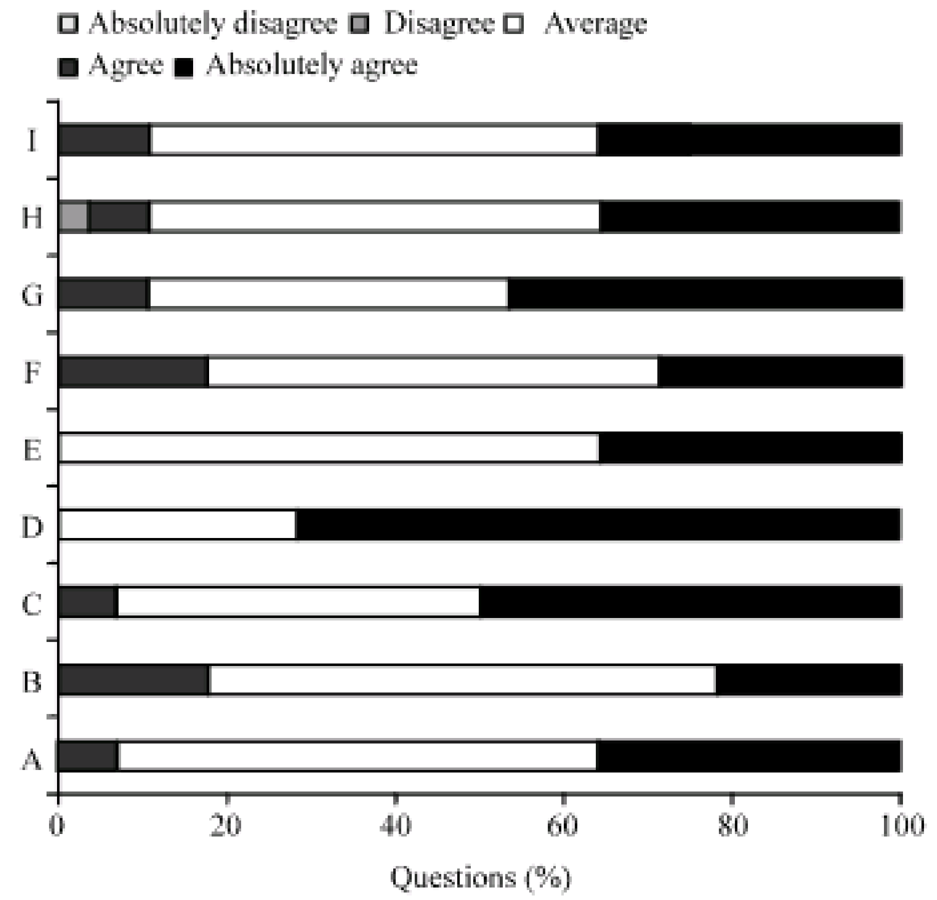

Fig. 6: Bar chart of students' opinions on system usability

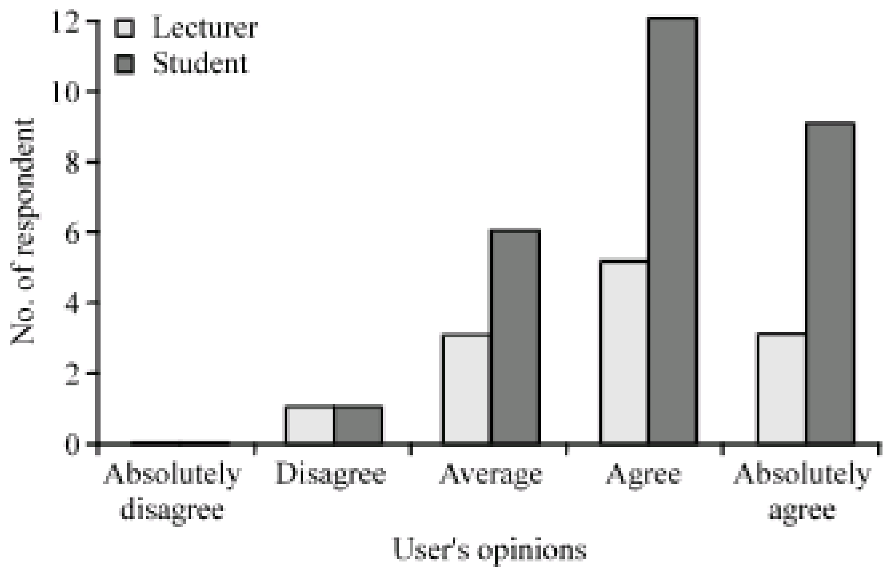

Fig. 7: Bar chart of users' opinions on the overall system

In the Fig. 5 and 6 show the results for lecturers' and students' opinions on the SAMS's interface. Most of them chose either agree or absolutely agree on the system usability. Therefore from the survey it is proven that the system fulfils overall usability requirements.

Figure 7 shows that 67% of lecturers and 75% of students were satisfied with the overall system and none of them choose absolutely disagree when it turns to SAMS overall satisfaction. As the conclusion, the system meets the functional requirements from the users' perspective.

CONCLUSION

The developed SAMS turns the traditional appointment management approach into a Web based system by applying Web Modeling Language (WebML) and .NET four-tier architecture so that students and lecturers can arrange meetings in an efficient way and to make better use of web infrastructure and connectivity. SAMS not only aims to fulfill these functional requirements, but is also designed to satisfy non-functional requirements, namely usability and maintainability. The Goal-Question-Metrics (GQM) approach was used to measure the usability and to determine the effectiveness of the interface. SAMS provides a simple and user-friendly interface. The users can perform their tasks easily. By applying. NET four-tier architecture, WebML and cyclomatic complexity technique, SAMS achieves maintainability. .NET four-tier architecture offers maximum functionality and flexibility in a heterogeneous web based environment, while WebML guarantees a model-driven approach, which can reduce development efforts (cost and time) and allows a more structured development process. It is proven through questionnaire findings that good usability of SAMS has been achieved as 67% of lecturers and 75% of students were satisfied with the overall system and none of them choose absolutely disagree when it turns to SAMS overall satisfaction.

ACKNOWLEDGMENTS

First and foremost we would like to express our gratitude to the Almighty who gave us the opportunity to complete the research successfully. Secondly, we would like to forward our deepest thanks to our colleagues, lecturers and technical staff from the Department of Software Engineering for their endless assistance, technical advice and co-operation.

REFERENCES

Behravanfar, R., 2001. Separation of data and presentation for the next generation internet using the four-tier architecture. Proceeding of the 39th International Conference and Exhibition on Technology of Object-Oriented Languages and Systems, Jun. 29 to Jul. 3, Santa Barbara, CA., USA., pp: 83-88.

Card, D.N. and W.W. Agresti, 1988. Measuring software design complexity. J. Syst. Software, 8: 185-197.

Ceri, S., P. Fraternali and M. Matera, 2002. Conceptual modeling of data-intensive web applications. IEEE Internet. Comput., 6: 20-30.

Ceri, S., F. Daniel, M. Matera and F.M. Facca, 2007. Model-driven development of context-aware web applications. ACM Trans. Internet Technol., 7: 2-2.

Fenton, N.E. and S.L. Pfleeger, 1998. Software Metrics: A Rigorous and Practical Approach. 2nd Edn. PWS Publishing, Boston, I SBN-10: 0534954251, pp: 320-336.

Furht, B., C. Phoenix, J. Yin and Z. Aganovic, 2000. An innovative internet architecture for application service providers. Proceedings of the 33rd Hawaii International Conference on System Sciences, Jan. 4-7, Boca Raton, Florida, pp: 1-10.

Ginige, A. and S. Murugesan, 2001. Web engineering: An introduction. IEEE, 8: 14-18.

Kim, K., Y. Shin and C. Wu, 1995. Complexity measures for object-oriented program based on the entropy. Proceedings of the 2nd Asia-Pacific Software Engineering Conference, Dec. 6-9, Brisbane, Qld., Australia, pp: 127-136.

Rosenberg, L.H. and L. Hyatt, 1998. Applying and interpreting object oriented metrics. Proceedings of the Software Technology Conference, Utah, April 1998.

Schewe, K.D. and B. Thalheim, 2005. Conceptual modelling of web information systems. Data Knowledge Eng., 54: 147-188.