Wireless Pers Commun (2013) 71:2161–2179
DOI 10.1007/s11277-012-0929-3

Design and Implementation of a Scalable RFID-Based Attendance System with an Intelligent Scheduling Technique

Mohammed I. Younis · Zinah Fadhil Abed Al-Tameemi · Widad Ismail · Kamal Z. Zamli

Published online: 13 November 2012

Abstract Radio Frequency Identification (RFID) technology; a convenient and flexible technology which is well suited for fully automated systems, is directing human lifestyle towards automation and reality. Integrating RFID into attendance management systems makes the tasks of both users and administrators easy, smart, convenient, and practical. Earlier implementations of RFID-based attendance systems involve different approaches and facilities. Different intertwined characteristics (i.e., scalability, and automation) are suggested in evaluating the performance of these systems. The coverage of these characteristics appears to be adequate in achieving a good systems implementation. Some of the existing systems present high performance; however, these systems still have gaps in some of the suggested characteristics. In order to mind these gaps, a scalable RFID-based attendance system with an intelligent scheduling technique; called Intelligent and fully Automated Attendance System (IAAS), has been designed and implemented. Finally, the performance of IAAS has been evaluated through a comparison with existing attendance systems.

Keywords RFID · Scalability · Automation · Load-balancing · Cross-platform · Scheduling

M. I. Younis (✉) · Z. F. A. Al-Tameemi
Computer Engineering Department, College of Engineering, University of Baghdad, Baghdad, Iraq
e-mail: younismi@gmail.com

Z. F. A. Al-Tameemi
e-mail: zinahfa@gmail.com

W. Ismail · K. Z. Zamli
School of Electrical and Electronics Engineering, Universiti Sains Malaysia, Penang, Malaysia
e-mail: eewidad@eng.usm.my

K. Z. Zamli
e-mail: eekamal@eng.usm.my

1 Introduction

Radio Frequency Identification (RFID) systems have been incorporated into a wide range of industrial and commercial systems [7]. RFID is an electronic technology that allows one to automatically identify and locate objects, people, and animals in a wide variety of deployment settings. In RFID, radio frequency signals are used to transfer data between two entities (i.e., a reader and a tag); hence, RFID is described as an automated data collection technology. An RFID system is an integrated collection of components that implement an RFID solution for a desired application. An RFID-based system serves the purposes of identifying, monitoring, authentication, and alerting through the exchange of data between tags and readers [21]. From an end-to-end perspective, an RFID system consists of RFID tag, RFID reader, reader antenna, and a back-end database [4,20].

RFID applications are growing rapidly and are receiving considerable worldwide attention. The applications of RFID include the identification of both people and products [13,26]. RFID systems have been successfully applied in many areas such as manufacturing, supply chains, agriculture, transportation, library, managing toxic and hazardous chemicals, and healthcare services [14,21,29]. One of these applications is the attendance tracking and tracing.

Automating work and realizing the real-time tracking of individual locations are popular incentives derived from attendance systems. Using such systems improves organizational efficiency and performance, and increases the overall satisfaction of individuals. Manual paperwork wastes effort, time, and money. Furthermore, many problems may occur with manual paperwork, including possible big mistakes due to huge data, and impossibility of remotely checking attendance status.

RFID technology offers low-cost and convenient object identification alternative in a physically contactless manner; hence, it is the most suitable base technology in developing attendance management systems [8]. However, the development of such systems face multiple challenges in terms of scalability and reliability, calling for the use of distributed system technologies [3]. Another challenge is developing RFID systems that can efficiently collect, filter, and save detected information. In short, RFID systems must be managed in a scalable fashion because they generate numerous events in the form of data streams in a short time [3]. In order to overcome all these challenges, an Intelligent and fully Automated Attendance System (IAAS) will be proposed.

The rest of this paper is organized as follows. Section 2 highlights the desired characteristics in developing RFID-based systems. Section 3 gives a comparison among existing attendance systems in the literatures. Section 4 introduces the architectural design of the proposed system. The implementation of the IAAS will be presented in Sect. 5. Section 6 contains an evaluation and discussion for the IAAS, and a comparison between the IAAS and the existing attendance systems. Finally, conclusion is presented in Sect. 7.

2 Desired Characteristics on Developing RFID-Based Systems

RFID labels play an important role as an inventory tracking technology [12]. In addition, RFID is considered much more secure compared with other networks [28]. Beyond simple identification, other functions, such as integrated detection, read/write storage, and access control, may be incorporated into RFID systems. This section is focusing on the required intertwined characteristics (i.e., scalability, and automation), as follows.

1. Scalability—It indicates the desirable ability of a system, a network, or a process to handle growing amounts of work in a smooth manner or be readily enlarged [1,5] Scalability issue includes the following features:
   (a) Centralized/Distributed: Centralized or distributed systems support the ability of placing the system on single or multiple machines and dealing with centralized or distributed databases [22]. In contrast, centralized services are more appropriated for small scale systems [19]. A trade off between the two mechanisms is also desired for a distributed management system [16].
   (b) Cross-platform is the feature allows software to run identically across different operating systems [2].
   (c) Load Balancing: is the ability to distribute the amount of work that a host has to do among two or more hosts so that more work gets done in the same amount of time and, in general, all users get served faster [9,10].
   (d) Highly Multithreaded: is the ability of running multiple tasks at the same time [2].
   (e) Remote and Concurrent Tracking: remote tracking is the ability of tracking the attendance situation remotely, for example browsing the generated reports using web [24]. Whilst concurrent tracking is the ability to track the attendance in different places simultaneously [2,18,19].
2. Automation: this issue includes the two following features:
   (a) Scheduling: real-time systems have to complete the execution of a task within the predetermined time. Such systems require scheduling methods that can adequately distribute the given tasks to a worker. Scheduling methods that all tasks can be executed within a predetermined deadline are called an optimal scheduling [15]. Scheduling manages multiple job streams across both single and multiple PCs with the objectives of improving the system utilization [6].
   (b) Automated Operations: the automated operations aim to minimize human interaction through: creation and updating of required jobs, start and termination of jobs, dealing with required acquisition processes, updating the database accordingly, generating real-time reports, and website updating in real time.

3 Related Work

Attendance management has been considered as an important organizational issue. However, fully automated attendance management is still not appropriately supported by existing attendance systems. The evaluation of each attendance system guides the proper analysis of the present works. Therefore, different attendance systems will be compared based on the desired characteristics defined in previous section.

The first automated attendance system to be evaluated is Automation of Time and Attendance using RFID Systems (ATAS) proposed by Qaiser and Khan [24]. The system is specialized for university environments, including students and faculty members. ATAS is an efficient automated system, which is helpful in automating student management. On the other hand, ATAS requires a dedicated PC to be attached to each RFID reader. This requirement significantly affects the scalability and the overall cost of the system. Although each PC records the attendance simultaneously, ATAS cannot concurrently track attendance, because the timer used by Windows Service updates the database every 24 hours. Furthermore, ATAS is dedicated to a specific system (i.e., Windows platform). This ultimately affects the scalability and reusability of the system. Finally, the acquisition processes of the system must be terminated by stopping the whole system.

Lim et al. proposed the RFID-Based Attendance System (RFIDBAS) [19]. The system can identify and take the attendance of persons in schools, colleges, universities, and any workplace. The RFIDBAS uses a serial port to connect the reader to the PC; hence, the scalability of the system is highly limited. Moreover, the use of the system is limited by the need to have an operating system providing the HyperTerminal software; hence, the system does not support cross platform functionality. The final issue involves the automation functionality of RFIDBAS, which is also limited. The only automated operations in RFIDBAS are the update and storage of detected information in the database.

Wahab et al. proposed the Portable Examination Attendance System (PEAS) [27]. PEAS is developed and implemented on a Personal Digital Assistant (PDA) as a small scale application to take the attendance of the students during the examination. PEAS; like other automated attendance systems, reduces time, manpower, long-term cost (of printing and paper), and eases the examination procedures. On the other hand, the scalability of the PEAS is very limited due to user centralized. Moreover, the automation is also limited. The only automated operations in PEAS are the update, storage of detected information in the database, and the generation of real time reports.

Ali et al. proposed a networked based RFID Tracking and Monitoring (RFIDTM) system [2]. RFIDTM system is developed using Java programming language and has been tested on many platforms. RFIDTM can be run in a centralized or distributed environment, provides cross platform, highly multithreaded, and remote and concurrent tracking features. In addition, RFIDTM provides the update, storage of detected information in the database, real time reports generation, and update the website automatically. On the other hand, RFIDTM does not provide load balancing feature, scheduling, creation and updating the acquisition processes. Furthermore, the starting and stopping of the acquisition processes are done manually by pressing ‘Start’ and ‘Stop’ buttons respectively.

Finally, Singhal and Gujral proposed a remote monitoring of Attendance System based on RFID using GSM Network (ASRFIDGSM) [25]. There is a lot of benefits of the system i.e. students attendance record to the parents on daily basis, employee’s attendance notification. Thus, ASRFIDGSM supports centralized management, remotely (via SMS) and concurrently tracking as far as the scalability feature is concerned. In addition, dealing with required acquisition processes, updating the database, and generating real-time reports are provided as far as the automation operations are concerned. On the other hand, other desired characteristics involve: cross platform, multithreading, load balancing, scheduling, job’s creation and updating, start and termination of jobs, and website updating are not supported in this system.

From the above analysis, it is clear that some common desired features are missing in all reviewed RFID-based attendance systems, involve :load balancing, scheduling, job’s creation and updating, and termination of jobs. In addition, each system has its merits and demerits as far as other features are concerned. Fix and build from the earlier works, the architectural design of the IAAS to tackle the desired characteristics will be discussed in the next section.

4 IAAS Architectural Design

The proposed attendance system is carefully designed to fulfil the requirements of a fully automated, scalable, and real-time attendance-tracking system. The overall design of the

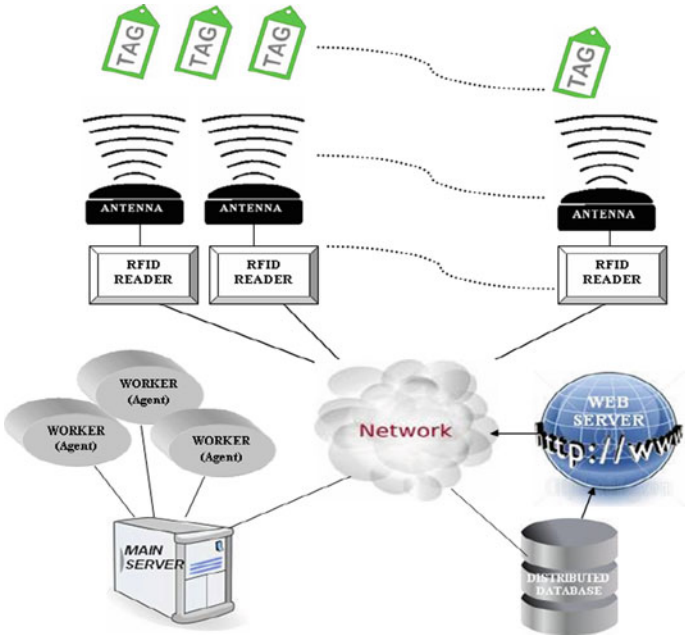


Fig. 1 IAAS architectural design

system is revealed and constructed depending on the hardware and software designs of the system.

The simplified design of the IAAS architecture is demonstrated in Fig. 1. IAAS consists of seven components listed and explained below.

1. Identification tags
2. Identification reader devices—In order to detect the tags entering or getting out from a specific place, the readers should be placed in suitable places. The readers then detect any tag within their working range, which varies according to the type of reader used.
3. Reliable network—All parts of the system are connected together by a network; hence, the performance of IAAS depends on the reliability of such a network. Wireless networks, LAN, or WAN may be used in IAAS. What solely matters is the network's reliability.
4. Server—In IAAS, the server is the main controller of all the system's operations. The server is represented by a PC with any operating system (e.g., Windows, MAC, or Linux).
5. Workers (agents)—The workers (agents) in IAAS are represented by threads created by multiple micro-processors in the same PC or by multiple computers distributed over a network of interconnected computers. The main function of a worker is to handle the job assigned by the server.
6. Distributed database—A database server is needed in IAAS for storing, scheduling, updating, and reporting all information related to the system. The database is distributed under the control of a central Database Management System (DBMS).
7. Web server—A web server is used to update the website related to the system in real time. The web server is directly connected to the corresponding database and executes real-time queries for real-time tracking.

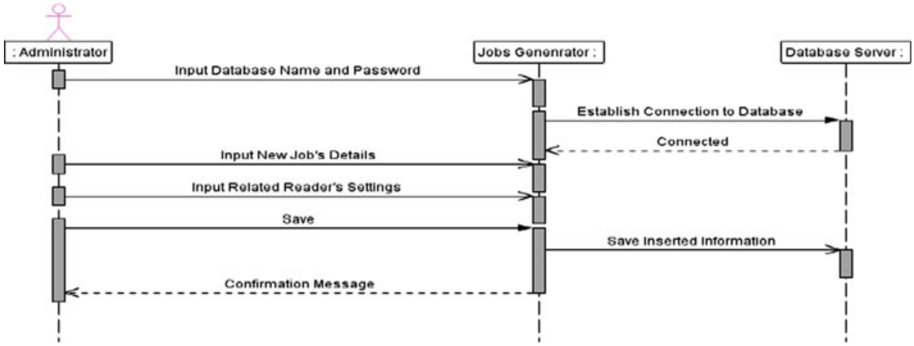


Fig. 2 Sequence diagram for creating new jobs

The software design of IAAS contains many modules.. In addition, sequence diagrams are used to demonstrate the processes flow inside the system. These modules are described as follows.

4.1 Initialization Module

The initialization module is the first stage in the system's operation. It interacts with two actors, namely, the administrator and the scheduler software installed in the main server. According to functionality, the initialization module can be separated into two parts, the first of which is the creation of new jobs. In order to make the designed system user friendly, without requiring a database and Java programming specialist, the whole creation of the database is automatically performed by the installed scheduler software. The scheduler software is responsible for creating a new database having all the required tables. All that is asked from the administrator is to input information required for the system's operation (e.g., name of the new database, password, name of new jobs, and information on the jobs). As so, the administrator can create new jobs by simply providing the software with the required information. On the other hand, the scheduler software handles the real creation of the database and the tables, including the creation and storage of schedulers for new jobs.

A sequence diagram is constructed to show all the classes and messages needed for this module as shown in Fig. 2.

4.2 Auto-Scheduling Module

The auto-scheduling module provides the system with fully automated operations. With this module, there is no need for daily human interaction in initiating and terminating the system's processes. The main actor in this module is the main server. The main functions described below are performed by the main server in the auto-scheduling module.

1. Annual checking of schedulers—For every Job, the stored scheduler is frequently checked to track the 'starting time'. With this function, there is no need for human interaction in initiating the attendance-tracking process for every stored job.
2. Load balancing—When a job is activated or when its start time begins, the main server checks the load balancing table to look for a free worker and assigns the job to it. Each worker runs in a separate thread, that created by available CPU that has a minimum load. If all CPUs have the same load, then the job will be assigned in a round-robin

Table 1 Example of load balancing mechanism

Number of created threads			Action of the information server
CPU#1	CPU#2	CPU#3	
2	4	1	Assign job to CPU#3
2	4	2	Assign job to CPU#1
3	4	2	Assign job to CPU#3
3	4	3	Assign job to CPU#1
4	4	3	Assign job to CPU#3
4	4	4	No more activated jobs

fashion. Initially, The information server creates an individual counter to count the jobs assigned to each CPU. After assigning a job to a specific CPU, the dedicated counter is increased by one. After the assigned job is accomplished, the dedicated counter is decreased by one, and so forth. For purposes of load balancing, these counters assist the information server in determining the load status of each connected CPU. For further explanation on the mechanism used for balancing the load, an example is shown in Table 1. In this example, we assume that the main server has five activated jobs to be assigned to the workers. In addition, there are three CPUs that handle these jobs: namely: CPU#1, CPU#2, and CPU#3. During the run time, CPU#1 has two active workers (i.e., status counter=2), CPU#2 has four active workers, and CPU#3 has one active worker. Since, CPU#3 has minimum work load, the information server assigns the first job to it (see first row of Table 1) and increments the status counter by one (see second row of Table 1). Next, both CPU#1 and CPU#3 have two active workers, the information server assigns job to CPU#1. After that to CPU#3, and so forth. By using this technique, load balancing is achieved (see last row of Table 1).

3. Assign jobs to available workers—When the 'start time' of a specific job begins, immediately, the main server looks for a free connected worker and assigns the job to it. Assigning the job also includes providing the worker with required information about the reader (i.e., the IP address and port number), the database name, and the period of the assigned job.
4. Specify the time for the operation-The most important information that the server provides the worker is the time of the operation. Specifying this period ensures the ability to terminate the attendance-tracking process automatically without any human intervention.

The sequence diagram for the auto-scheduling module is shown in Fig. 3.

4.3 Attendance-Tracking Module

This module contains two parts: Data Detection Process (DDP) and Data Acquisition Process (DAP).

Data Detection Process (DDP): When the attendance-tracking process begins, any tag within the range of the working reader is detected and all its information is transferred to the worker assigned to the job. The sequence diagram for DDP is shown in Fig. 4.

Data Acquisition Process (DAP): The worker assigned to the job is the only actor in the DAP module. The functions performed by the worker in this module are presented below.

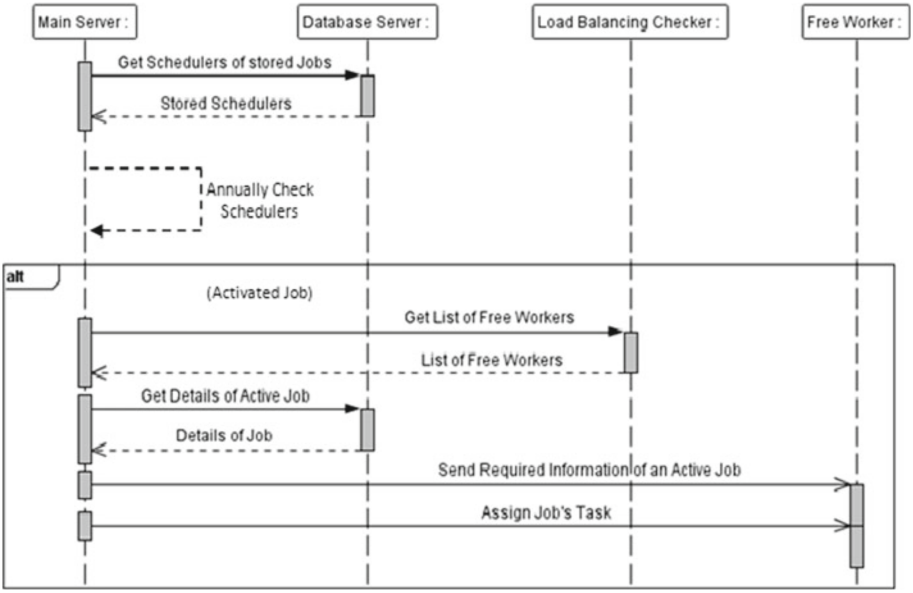


Fig. 3 Sequence diagram for the auto-scheduling module

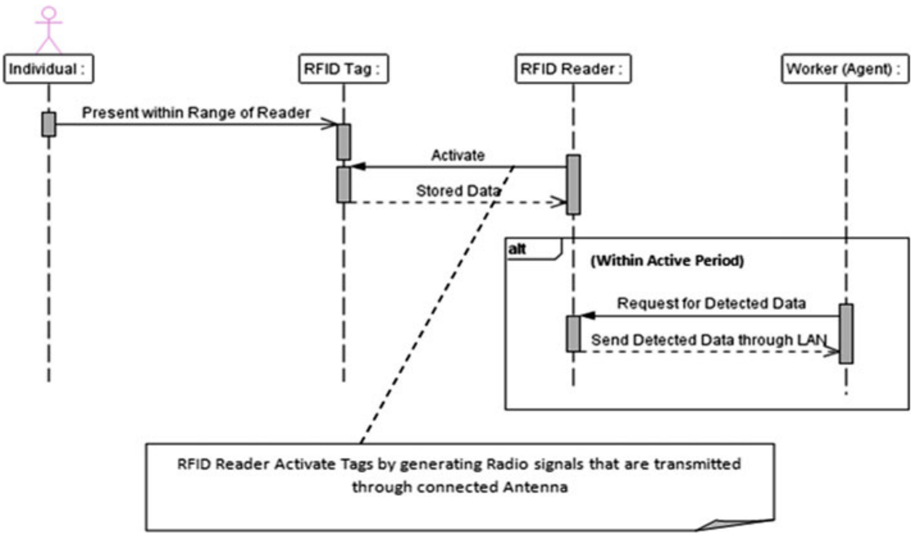


Fig. 4 Sequence diagram for the DDP module

1. Mark a "free" situation—Here; the free workers are the workers (CPUs) without jobs or with low loads so that they can handle any job assigned by the server. It is necessary for the main server to identify the free workers; hence, whenever a worker is free, a free "flag" situation is created. Every worker (CPU) creates a dedicated thread for every assigned job.

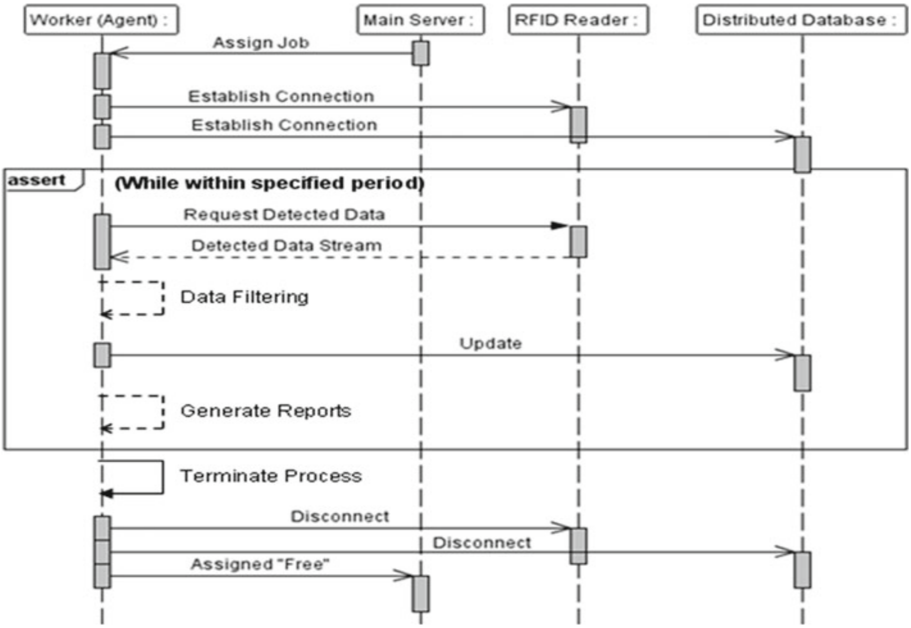


Fig. 5 Sequence diagram for the DAP module

2. Obtain a job—When the worker is free, it waits to be assigned by the main server to handle a job.
3. Connect to the specified reader and database—For every stored job, there is a specific reader with specific connection settings and specific related database information. After obtaining information from the main server, the worker establishes correct connections to both the reader and the database.
4. Receive data—The acquisition process includes detecting and filtering dataflow sent by the reader.
5. Update the database—After filtering; the revealed information is stored in the corresponding database for tracking purposes. Automatically, different types of reports are generated in real time for any tracker using the system.
6. Terminate the process automatically—One of the received information from the server is the time period specified for the job. As soon as this period elapses, the worker immediately terminates the attendance-tracking process and returns to its original state (i.e., free state).

The detailed information about the processes performed in this model is shown in the sequence diagram in Fig. 5.

4.4 Remote Tracking Module

With this module, the remote tracking feature can be achieved using Web service. The web server continuously updates the website in order to track the attendance situation in real time. After establishing a connection to the corresponding database, all reports displayed on the website are updated to follow the real state of the attendance situation. Thereafter, any interested user can follow the real-time state of the attendance by simply browsing the

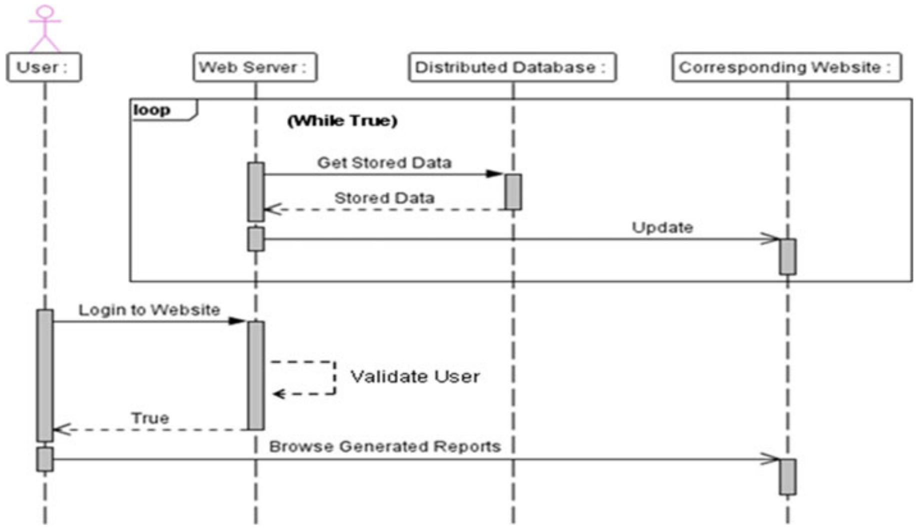


Fig. 6 Sequence diagram for the remote tracking module

website. For security purposes, every user is supplied with a login username and password to gain access to the website.

The sequence diagram for the remote tracking module is shown in Fig. 6.

5 IAAS Implementation

For the IAAS implementation, constructed Java classes and GUIs are classified into three modules. These are discussed in detail below.

5.1 Administration Module

The administration module is responsible for the initialization, update, and auto generation of schedules used by the system. Full authority is given to the administrator to add, delete, or update as many jobs as desired. The constructed GUI for the administration module (Fig. 7) facilitates the creation of a new database for every new group of jobs. Daily schedules for each stored job are generated automatically to ensure the automatic initialization and termination for every acquisition and database updating processes.

5.2 Acquisition Module (Controller and Executers)

The acquisition process includes detecting, receiving, and filtering dataflow received from the detecting device (reader). The acquisition process mainly consists of two parts: the controller and executers. Referring to the class diagram depicted in Fig. 8, the following points can be noticed:

1. The runner-acquisition class is the main controller and motivator in the acquisition module;
2. The runner-acquisition class annually checks the stored schedules and activates jobs accordingly;

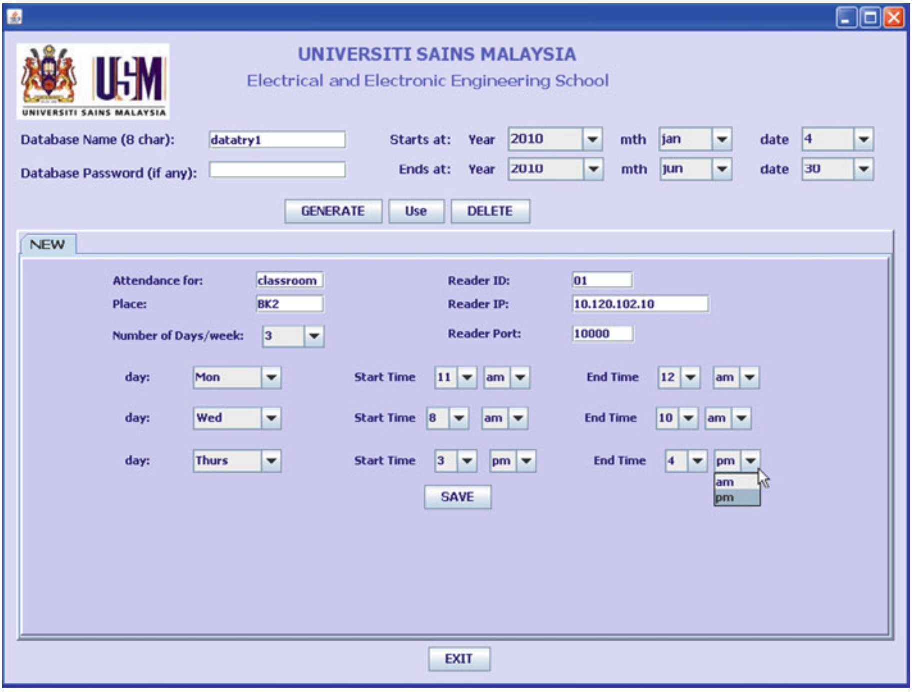


Fig. 7 GUI for administration module

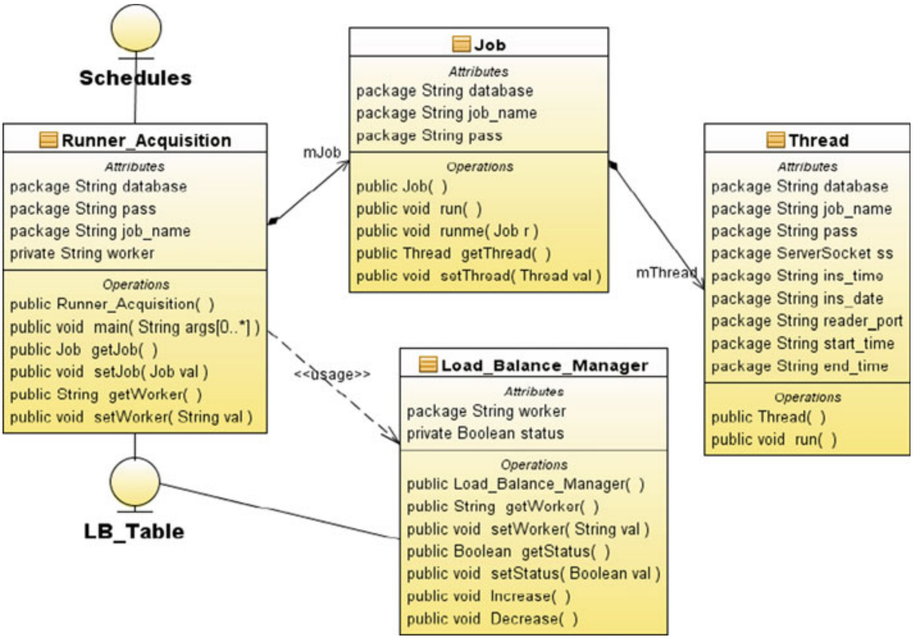


Fig. 8 Class diagram for the acquisition module

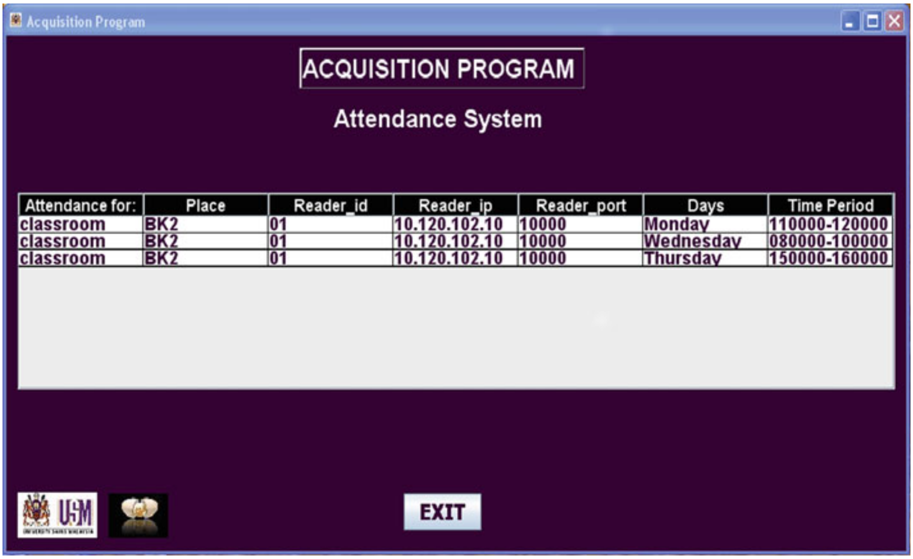


Fig. 9 Acquisition controller program

3. The runner-acquisition class checks the load balancing table for free workers to be assigned with the activated job;
4. When a job is activated, the runner-acquisition class creates a new class; Job, to be handled by the worker; and
5. The worker creates a thread class to handle the connection to the reader and database, acquisition process, and database updating operations.

The controller in the acquisition module is represented by the GUI shown in Fig. 9. The controller GUI displays the jobs being executed and all the related information for monitoring purpose. The main function of the controller GUI is to perform load balancing by assigning jobs within a specified period to free workers. Furthermore, using the acquisition controller, automation is highly achieved by auto-handling required jobs with no need for any human interaction. When the job begins, the executing GUI is activated to display all acquired data after receiving it from the reader (Fig. 10).

Two modules of the architectural design are covered by implementing the acquisition module, namely: auto-scheduling module and attendance-tracking module (including its two parts, DDP and DAP).

5.3 Reports Generators and Website Browser Module

The constructed GUI for selecting the job and the type of report to be displayed is shown in Fig. 11. When a specific date is selected, attendees, absentees, and foreign identities are displayed along with their corresponding collected information (Fig. 12). In addition, these reports can be printed if desired.

The report generation system was implemented to support internet browsing. The web browsing is implemented using Java Server Page (JSP) technology, which provides a simplified and rapid way of creating dynamic web content. This JSP technology enables the rapid development of web-based applications that are server—and platform-independent [23]. Fur-

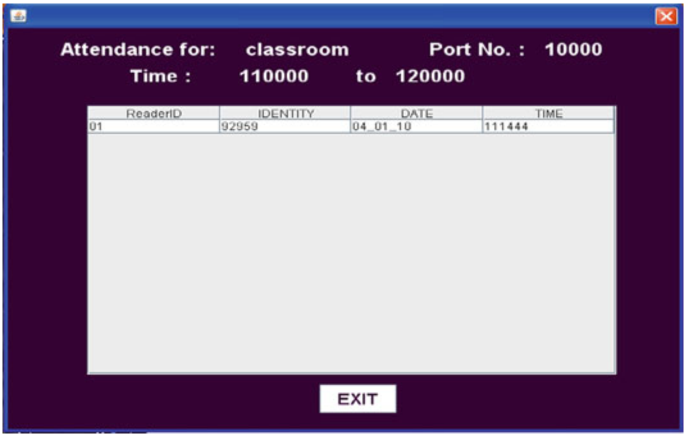


Fig. 10 Activated acquisition thread

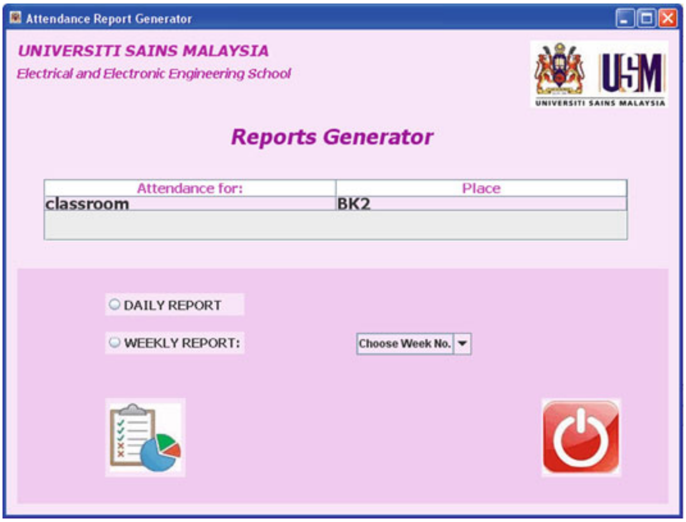
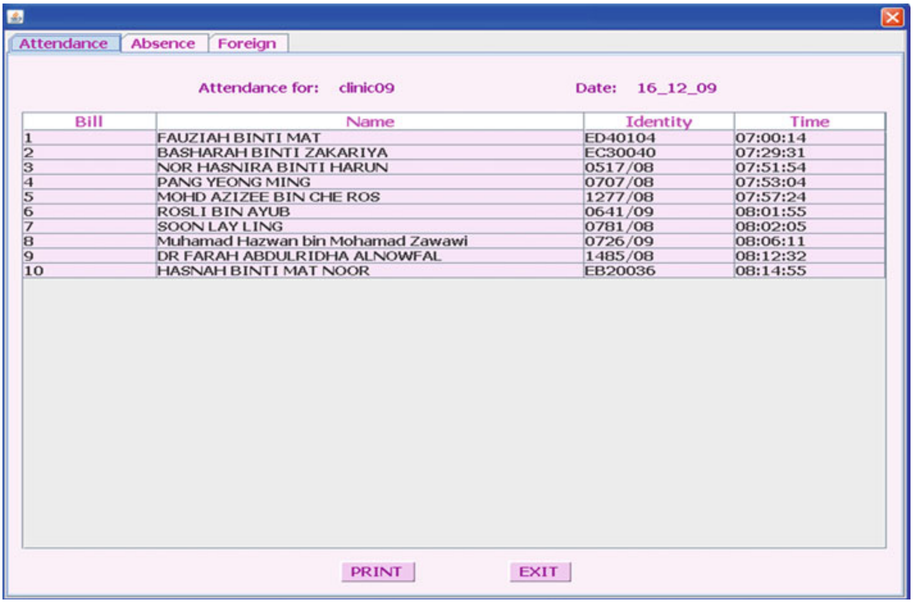


Fig. 11 Main utility programs for reports generation

thermore, with web browsing using the JSP feature, any user can browse the reports of IAAS without any restriction on the used OS, because it can be operated on different operating systems (e.g., Windows, MAC, Linux) as can be seen in Fig. 13.

6 Evaluation and Discussion

In this section, the performance of the proposed system will be discussed and evaluated in terms of the required characteristics as far as the scalability, and automation are concerned. A comparison among all attendance systems will be presented in terms of a tabular checklist.



Bill	Name	Identity	Time
1	FAUZIAH BINTI MAT	ED40104	07:00:14
2	BASHARAH BINTI ZAKARIYA	EC30040	07:29:31
3	NOR HASNIRA BINTI HARUN	0517/08	07:51:54
4	PANG YEONG MING	0707/08	07:53:04
5	MOHD AZIZEE BIN CHE ROS	1277/08	07:57:24
6	ROSLI BIN AYUB	0641/09	08:01:55
7	SOON LAY LING	0781/08	08:02:05
8	Muhamad Hazwan bin Mohamad Zawawi	0726/09	08:06:11
9	DR FARAH ABDULRIDHA ALNOWFAL	1485/08	08:12:32
10	HASNAH BINTI MAT NOOR	EB20036	08:14:55

Fig. 12 Attendees of the selected job and date

Fig. 13 Web browsing in different operating systems

6.1 Scalability Justification

Scalability in the IAAS design and implementation is represented by five features presented below.

1. Centralized or distributed—According to the application demand, the IAAS software can be installed and executed in a single centralized machine, including the software of the initialization, acquisition and reporting modules, or distributed in tightly (i.e., a single PC) or loosely coupled machines (i.e., PCs connected together through a network). The same holds true for the system's database, which can be centralized or distributed locally or via a web host. Furthermore, the required servers for the system (information, database, and web servers) can be in a single machine or separated in different PCs. This feature makes the overall system scalable: it can be laid in a single PC, or it can be distributed in loosely coupled machines. As such, like RFIDTM, unlike RFIDBAS, PEAS, and ASRFIDGSM that support centralised operations only, and unlike ATAS that supports distributed operations only, IAAS supports both centralised and distributed functionalities.

2. Cross-platform functionality—Like RFIDTM,unlike all other existing attendance systems
(ATAS, RFIDBAS, PEAS, and ASRFIDGSM), IAAS is implemented using pure Java coding; hence, the system provides cross-platform functionality.
3. Load balancing—Load balancing is accomplished by the information server by assigning jobs to workers. Load balancing ensures the capability of dealing with as many jobs as desired, thereby ensuring the ability of implementing the system in large-scale applications. Unlike ATAS, RFIDBAS, PEAS, RFIDTM, and ASRFIDGSM, IAAS supports load balancing feature.
4. Highly multithreaded—Like RFIDTM,unlike all other existing attendance systems (ATAS, RFIDBAS, PEAS, and ASRFIDGSM), IAAS is highly multi-threaded, enabling the information server to assign more than one job to the same PC (e.g., make acquisition for many readers simultaneously). As such, it better utilizes the computing power of the PC.
5. Remote and concurrent tracking—Like RFIDTM and ASRFIDGSM, unlike ATAS that support remote tracking only, unlike RFIDBAS that supports concurrent tracking only, and unlike PEAS, IAAS supports both remote and concurrent tracking of attendance situations. The real-time generated reports can be browsed using a dedicated website.

6.2 Automation Justification

For IAAS, automation requirements do not only include the automated dealing with RFID equipment and Java/MySQL commands, although these facilities are very important for non-professionals. The most significant part in the automation requirement is scheduling the automated initialization and termination of every acquisition process. The points explained below represent the major achievements of the automation feature in the IAAS design and implementation.

6. Scheduling—With the scheduling approach, system automation is efficiently improved, because there is no need for daily human interaction to initialize and terminate the acquisition processes. In any TCP/IP-based system, network failure is a common problem, which occurs due to overloading in case of continuous acquisition requests from every connected reader in the network. It is an evident that although overloads do not happen frequently, the failure resulting from them can be quite expensive [11,17]. Using the scheduling technique reduces the probability of network failure; this is because every reader has a specific date and period to deal with. Moreover, by using the scheduling technique, load balancing is achieved by the information server; hence, the scheduling technique is an intertwined factor for both automation and scalability. Furthermore, the absence of the need for daily human interaction reduces the required efforts to the minimum level, leading to big savings in time and cost. Unlike all reviewed attendance systems (ATAS, RFIDBAS, PEAS, RFIDTM, and ASRFIDGSM), IAAS provides scheduling feature.
7. Automated operations—In IAAS, all of the following operations are automatically performed without requiring professionals who are adept in Java programming or MySQL commands:

   i. Creation and updating of required jobs: unlike all reviewed attendance systems (ATAS, RFIDBAS, PEAS, RFIDTM, and ASRFIDGSM), this operation is automated in IAAS;

Table 2 Comparison between IAAS and Related RFID-based Attendance Systems

Characteristics	ATAS	RFIDBAS	PEAS	RFIDTM	ASRFIDGSM	IAAS
Scalability						
Centralized/distributed	Only distributed	Only centralized	Only centralized	√	Only centralized	√
Cross-platform functionality	×	×	×	√	×	√
Load balancing	×	×	×	×	×	√
Highly multithreaded	×	×	×	√	×	√
Remote and concurrent tracking	Only remote tracking	Only concurrent tracking	×	√	√	√
Automation						
Scheduling	×	×	×	×	×	√
Automated operations						
Jobs' creation and updating	×	×	×	×	×	√
Start of jobs	√	×	×	×	×	√
Termination of jobs	×	×	×	×	×	√
Dealing with required acquisition processes	√	√	√	√	√	√
Updating the database	√	√	√	√	√	√
Generating real-time reports	×	×	√	√	√	√
Website updating in real time	×	×	×	√	×	√

ii. Start of jobs: like ATAS, unlike RFIDBAS, PEAS, RFIDTM, and ASRFIDGSM, IAAS supports automation for starting jobs;

iii. Termination of jobs: unlike all reviewed RFID-based attendance systems (ATAS, RFIDBAS, PEAS, RFIDTM, and ASRFIDGSM), IAAS provides automatic termination of jobs;

iv. Dealing with required acquisition processes: like any RFID-based attendance systems, this operation is automated in IAAS;

v. Updating the database accordingly: like any RFID-based attendance systems, this operation is automated in IAAS;

vi. Generating real-time reports: like PEAS, RFIDTM, and ASRFIDGSM, and unlike ATAS, and RFIDBAS, IAAS generates reports automatically in real time; and

vii. Website updating in real time: like RFIDTM, and unlike ATAS, RFIDBAS, PEAS, and ASRFIDGSM, the website is updated in real time.

In order to summarize the performance of the proposed system comparing with the existing attendance system, a tabular checklist is presented in Table 2. According to Table 2, the shaded cells present the unique characteristics which are distinguished the proposed IAAS among other RFID-based attendance systems.

7 Conclusion

This paper presented a fully scalable, automated, and reusable RFID system called IAAS. The work investigated existing RFID-based attendance systems, in which the investigations involved evaluating the performance of these systems in terms of elected characteristics. Fix and build from earlier works, IAAS supports auto load balancing feature integrated with auto scheduling technique. In doing so, there is no need for daily human interaction. Saving efforts, time, and costs needed in the daily initialization and termination of the system is achieved by using schedules that are automatically generated by the system.

References

1. Ahn, S., Ryu, W., Hong, B., Chae, H. & Lee, J. (2010). Dynamic thread management for scalable RFID middleware. In *Proceedings of the international conference on information science and applications* (ICISA 2010), p. 1–8, Seoul, South Korea.
2. Ali, M. F. M., Younis, M. I., Zamli, K. Z., & Ismail, W. (2010). Development of Java based RFID application programmable interface for heterogeneous RFID system. *The Journal of Systems and Software, 83*(11), 2322–2331.
3. Anagnostopoulos, A., Soldatos, J., & Michalakos, S. G. (2009). REFiLL: A lightweight programmable middleware platform for cost effective RFID application development. *Pervasive and Mobile Computing, 5*(1), 49–63.
4. Balitanas, M. O., & Kim, T. (2010). Review: Security threats for RFID-sensor network anti-collision protocol. *International Journal of Smart Home, 4*(1), 23–36.
5. Bondi, A. B. (2000). Characteristics of scalability and their impact on performance. In *Proceedings of the 2nd international workshop on software and performance*, pp. 195–203, Ottawa, Ontario, Canada.
6. Bubak, M., Albada, G. D. V., Sloot, P. M. A., Dongarra, J. J., & Abawajy, J. H. (2004). Dynamic parallel job scheduling in multi-cluster computing systems. *LNCS, 3036*, 27–34.
7. Chen, M., Gonzalez, S., Zhang, Q., Li, M., & Leung, V. (2010). A 2G-RFID based E-healthcare system. *IEEE Wireless Communications Magazine, 17*(1), 37–43.
8. Chien, H.-Y., & Chen, C.-H. (2007). Mutual authentication protocol for RFID conforming to EPC class 1 generation 2 standards. *Computer Standards & Interfaces, 29*(2), 254–259.
9. Cui, J. F., & Chae, H. S. (2007). Developing load balancing system for RFID middlewares using mobile agent technology. *Agent and Multi-Agent Systems: Technologies and Applications, LNCS, 4496*, 757–764.
10. Dong, Q., Shukla, A., Shrivastava, V., Agrawal, D., Banerjee, S., & Kar, K. (2008). Load balancing in large-scale RFID systems. *Computer Networks: The International Journal of Computer and Telecommunications Networking, 52*(9), 1782–1796.
11. Garousi, V., Briand, L. C., & Labiche, Y. (2008). Traffic-aware stress testing of distributed real-time systems based on UML models using genetic algorithms. *Journal of Systems and Software, 81*((2), 161–185.
12. Goodruma, P. M., Mclarenb, M. A., & Durfeec, A. (2006). The application of active radio frequency identification technology for tool tracking on construction job sites. *Automation in Construction, 15*(3), 292–302.
13. Higgins, L. N., & Cairney, T. (2006). RFID opportunities and risks. *Journal of Corporate Accounting & Finance, 17*(5), 51–57.
14. Hwa, K. Y., Wu, M.-C., Yeo, S.-S., & Li, W. (2011). A GHS-based intelligent management platform with IP-camera security system. *Wireless Personal Communications, 56*(1), 85–96.
15. Hwang, M., Kim, P. & Choi, D. (2010). Least slack time rate first: New scheduling algorithm for multi-processor environment. In *Proceedings of the international conference on complex, intelligent and software intensive systems* (CISIS), pp. 806–811, Krakow.
16. Kloch, C., Kristensen, J. E., & Bilstrup, B. (2010). Future scenarios: What are the future services and applications?. *Wireless Personal Communications, 53*(3), 315–327.
17. Kuhn, R. (1997). Sources of failure in the public switched telephone network. *IEEE Computer, 30*, 31–36.
18. Lien, Y.-H., Hsi, C.-T., Leng, X., Chiu, J.-H., & Chang, K.-C. H. (2012). An RFID based multi-batch supply chain systems. *Wireless Personal Communications, 63*(2), 393–413.

19. Lim, T. S., Sim, S. C. & Mansor, M. M. (2009). RFID based attendance system. In *Proceedings of the IEEE conference of industrial electronics & applications* (ISIEA 2009), pp. 778–782, Kuala Lumpur, Malaysia.
20. Meng, S., Chen, W., Liu, G., Wang, S. & Wenyin, L. (2008). An asset management system based on RFID, webGIS and SMS. In *Proceedings of the 2nd international conference on ubiquitous information management and communication*, pp. 82–86, Suwon, Korea.
21. Nambiar, A. N. (2009). RFID technology: A review of its applications. In *Proceedings of the world congress on engineering and computer science*, pp. 1253–1259, San Francisco, USA.
22. Nou, R., Julia, F., Hogan, K., & Torres, J. (2011). A path to achieving a self-managed grid middleware. *Future Generation Computer Systems, 27*(1), 10–19.
23. ORACLE. (2012). Java Server Page Technology (JSP). Available at: http://www.oracle.com/technetwork/java/javaee/jsp/index.html.
24. Qaiser, A. & Khan, S. A. (2006). Automation of time and attendance using RFID systems. In *Proceedings of the 2nd international conference on emerging technologies* (IEEE-ICET 2006), pp. 60–63, Peshawar, Pakistan.
25. Singhal, Z., & Gurjal, R. K. (2012). Anytime anywhere- remote monitoring of attendance system based on RFID using GSM network. *International Journal of Computer Applications, 39*(3), 37–41.
26. Tofetgaard, T. S. (2010). Next generation wireless infrastructure. *Wireless Personal Communications, 53*(3), 465–479.
27. Wahab, M. H. A., Mutalib, A. A., Kadir, H. A., & Mohsin, M. F. M. (2010). Design and development of portable RFID for attendance system. In *Proceedings of the international conference on information retrieval & knowledge management*, pp. 173–178, Shah Alam, Malaysia.
28. Xiao, Y., Yu, S., Wu, K., Ni, Q., Janecek, C., & Nordstad, J. (2007). Radio frequency identification: technologies, applications, and research issues: Research articles. *Wireless Communications & Mobile Computing, 7*(4), 457–472.
29. Zhou, G., Jiang, P. & Zheng, M. (2008). Design of an RFID based manufacturing data tracking system in plant production. In *Proceedings of the 1st international conference on intelligent robotics and applications* (ICIRA 2008), pp. 688–696, Wuhan, China.

Author Biographies

Mohammed I. Younis obtained his BSc in computer engineering from the University of Baghdad in 1997, his MSc degree from the same university in 2001, and his PhD degree from the School of Electrical and Electronics Engineering, USM, Malaysia in 2011. He is currently a Senior Lecturer and a Cisco instructorat at the Computer Engineering Department, College of Engineering, University of Baghdad. He is also a software-testing expert in the Malaysian Software Engineering Interest Group (MySEIG). His research interests include software engineering, parallel and distributed computing, algorithm design, RFID, networking, and security. Dr. Younis is also a member of the Iraqi Union of Engineers, IEEE, IAENG, IACSIT, and IJCTE.

Zinah Fadhil Abed Al-Tameemi obtained her BSc degree in Computer Engineering from Al-Nahrain University in 2001, and her MSc degree from the School of the Electrical and Electronics Engineering, USM, Malaysia in 2011. She is also a member in the Malaysian Software Engineering Interest Group (MySEIG), and of the Iraqi Union of Engineers. Her research interests include software engineering, and RFID development.

Widad Ismail graduated from University of Huddersfield, UK in1999 and earned First Class Honors in Electronics and Communications Engineering and she received her PhD in Electronics Engineering from University of Birmingham, UK in 2004. She is currently an Associate Professor at the School of Electrical and Electronics Engineering, USM in NibongTebal, Penang, Malaysia. She has contributed extensively in research and in the areas of Radio Frequency Identification (RFID), Active Integrated Antennas(AIA), RF systems and Wireless Systems Design. She has initiated Auto-ID Laboratory (AIDL), Malaysiain 2008 as a research and commercialize oriented centre where the main objective is to become a hub for research and commercialization activities.

Kamal Zuhairi Zamli obtained his BSc in Electrical Engineering from Worcester Polytechnic Institute, Worcester, USA in1992, MSc in Real Time Software Engineering from CASE, Universiti Teknologi Malaysiain 2000, and PhD in Software Engineering from the University of Newcastle upon Tyne, UK in 2003. He is currently an Associate Professor attached to the Software Engineering Research Group, in the School of Electrical and Electronics Engineering, USM. His research interests include software engineering, software testing automation, and algorithm design.