DOI: 10.24996/ijs.2020.61.5.26

ISSN: 0067-2904

Development of an Attendance System Based on Cloud / Fog Computing with Data Recovery Capability

Mohammed I. Younis*[1], Manal F. Younis[1], Marwa M. Abed[1], AbdulRahman Ahmed Alsewari[2]

[1]Computer Engineering Department, University of Baghdad, Baghdad, Iraq
[2]Faculty of Computer Systems & Software Engineering, Universiti Malaysia Pahang, Pahang, Malaysia

Received: 28/7/ 2019 Accepted: 30/9/2019

Abstract

Given the high importance of attendance for university students, upon which the possibility of keeping or losing their places in the course is based, it is essential to replace the inefficient manual method of attendance recording with a more efficient one. To handle this problem, technology must be introduced into this process. This paper aims to propose an automatic attendance system based on passive Radio Frequency Identification (RFID), fog, and cloud computing technologies (AASCF). The system has three sides. The first one, which is the Client-side; works on collecting the attendance data then sending a copy from it. The second side, which is the Server-side, works on calculating an absence ratio of all the students during the course. The third side is the Fog-server. Data sent by the client-side reaches to the Fog-server which, in turn, sends data to the cloud at the end of the of working time at the university. This paper also reviews the state-of-the-art automatic attendance systems and shows the merits and demerits for each approach by providing a checklist comparison. Unlike the previous works, the proposed system protects data from wasting and ensures its arrival to the cloud even in cases of connection losing or device crashing, which is the contribution of this paper.

Keywords: Cloud computing, Attendance, Fog computing, RFID, Raspberry Pi
.

تطوير نظام الحضور المستند على الحوسبة السحابية / الضبابية مع إمكانية استعادة البيانات

محمد عصام يونس*[1] ، منال فاضل يونس[1] ، مروة محمد عبد[1] ،عبد الرحمن أحمد السيواري[2]

[1]قسم هندسةالحاسبات، جامعة بغداد، بغداد، العراق
[2]كلية نظم الحاسوب وهندسة البرمجيات ، جامعة باهانج الماليزية ، باهانج ، ماليزيا

الخلاصة

نظرًا لأهمية الحضور في الجامعة يعتمد عليه يتم تحديد أمكانية الطالب الاحتفاظ بمكانه في المقاعد الدراسية أو فقدها. لذلك ، فمن غير الفعال تسجيل الحضور يدويًا ،لأن ذلك سيؤدي إلى زيادة احتمال حدوث خطأ بالإضافة إلى إضاعة الوقت. لمعالجة هذه المشكلة، يجب إدخال التكنولوجيا في هذه العملية. الهدف من هذا البحث اقتراح نظام الحضور التلقائي على أساس تقانة التعريف الراديوية، والضباب ،والحوسبة السحابية. النظام له ثلاث جوانب. الجانب الأول ، وهو جانب العميل ، يعمل على جمع الحضور ثم إرسال نسخة منه. من الجانب الثاني ، وهو جانب الخادم، يقوم باحتساب نسبة غياب لكل الطالب. الجانب الثالث هو الخدام

*Email:younismi@gmail.com

الضبابي، حيث تصل البيانات التي يرسلها العميل إلى خادم الضباب الذي يقوم بدوره بإرسالها إلى السحابة في نهاية وقت العمل في الجامعة. لذلك يحمي هذا النظام البيانات من الضياع ويضمن وصولها إلى السحابة حتى في حالة فقد الاتصال أو تعطل الجهاز.

تستعرض هذه الورقة أنظمة الحضور الأوتوماتيكي الحديثة وتوضح مزايا وعيوب كل نهج من خلال توفير مقارنة قائمة مرجعية. بخلاف الأعمال السابقة ، فإن النظام المقترح يحمي البيانات من الضياع ويضمن وصولها إلى السحابة وحتى استرجاعها في حالة فقدان الاتصال أو تعطل الخادم الضبتبي وهو ما يمثل مساهمة هذه الورقة بالمقارنة مع الاعمال السابقة.

Introduction

Attendance systems are the core of institutions and universities; however, attendances are manually recorded by the tutor. Thus, these papers based records are prone to personal errors and are time-ineffective. There arises a need for effective and efficient methods to tackle these problems [1]. Moreover, the advancement in technologies makes a vital leap towards embedded systems and automation with low cost that facilitates the development of new tools. Hence, such advantages can be achieved using technologies such as low-cost mini-computer (Raspberry Pi) and cheap passive radio frequency identification (RFID) based on cloud and fog computing for developing such systems in a way that ensures data availability and safety. These technologies are briefly identified here.

RFID-based systems are a set of technologies that provide an ability to discover thing's location, whether it was a human, an animal or even an object. An RFID-based system facilitates the automation of identification and consists of four parts: an RFID reader, a tag, an antenna, and a database back-end server. The RFID reader sends radio waves to find the transponder, then it works on decoding the transponder integrated circuit. The transponder, which is simply a tag, has a large number of bits/signatures that is unique in each tag. This offers the ability to recognize things by their specific id number, that the tag contains. The tag signals are received by readers using an RFID antenna that consists of a coil with windings and a matching network. Finally, the received tag id will be saved in a back-end database [2,3].

Raspberry Pi is considered as a tiny and cheap PC that consists of a CPU, memory, and an SD card slot which is considered as the storage part and contains the operating system of Raspberry Pi, such as Raspbian. Raspbian is an open-source Linux OS made especially for Raspberry Pi. Furthermore, Raspberry Pi has GPIO pins allowing to integrate a touchscreen (LCD), as well as an HDMI four USB ports and an Ethernet port [4,5].

Cloud computing was created to provide services to the customer anytime and anywhere based on pay-as-you-go. Cloud computing allows users to access computing resources such as applications, storage, networking, and servers. These cloud's services are provided to the users by cloud service providers such as Google, Microsoft Azure, Amazon Web Services (AWS), etc. Cloud computing has two categorizations, the first is based on location and the other on services offered. The location categorization depends on the type of clients for whom the cloud is available, which is classified into four types [6], namely: public cloud, private cloud, community cloud, and hybrid cloud. Public cloud: is a low cost as compared to other cloud types and can be reached by everyone, thus it is the less secure cloud type. Private cloud is a higher cost and is available for a particular user or organization, hence it is the most secure type. Community cloud shares a database in addition to management between more than one organization. Hybrid cloud consists of a combination of other cloud types.

For the services offering category, there are four services [7], namely Recovery as a Service (RaaS), Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS), as depicted in Figure-1.

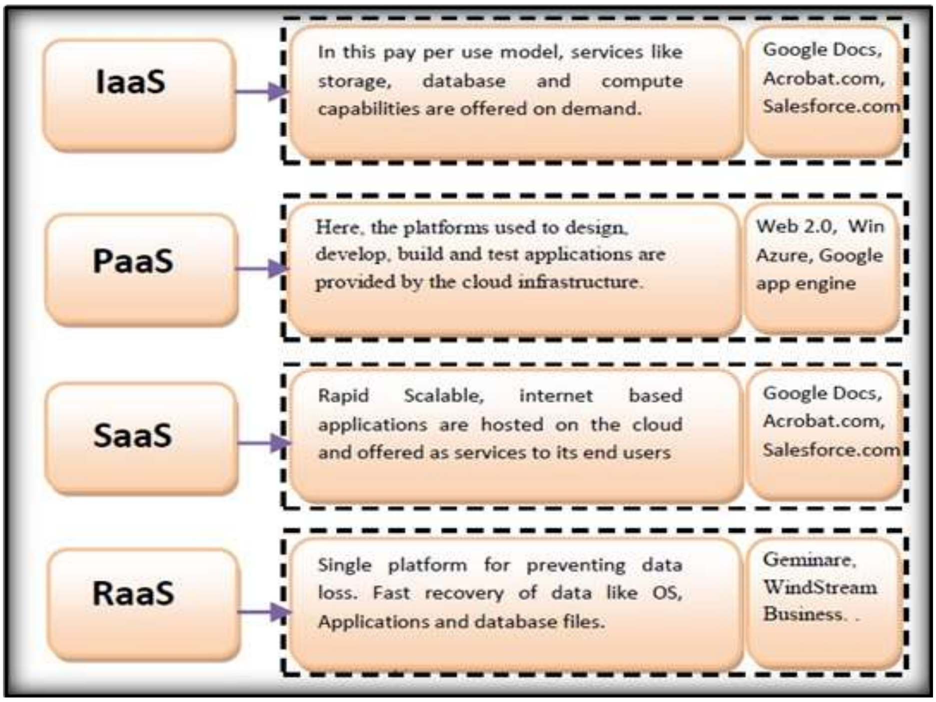

Figure 1- Service offering by cloud models [7].

Fog computing (or Fogging) was founded by Cisco Systems as a connection point located between cloud computing and clients. It provides closer computing and resources to the user than do the cloud, in addition to a wider geographical distribution. Moreover, it is useful to reduce the connection load to cloud computing which is required to increase efficiency and security. With fog computing, it is possible to address some cloud problems, such as the latency of data, lack of mobility support, and awareness of the site [8,9].

The rest of the paper reviews the state-of-the-art automatic attendance systems and identifies the strength as well as weakness features for each system. Based on the literature review, this paper proposes an Automatic Attendance System based on Cloud and Fog Computing (AASCF). The architectural design and implementation for the AASCF are discussed, followed by a discussion to evaluate the AASCF and to make comparisons with the related works. Finally, the conclusion and the direction for future works are stated.

Related Works

Due to the role that the attendance system plays in our lives, it becomes a focal point for the researchers. Different kinds of methods are being used to handle attendance technologically. The state-of-the-art research directions are discussed here.

Younis et al [2] proposed a scalable attendance system based on passive RFID technology. This Intelligent and fully Automated Attendance System (IAAS) has an artificial scheduling technique for power saving to turn on the reader only during the desired time of the lecturer. Also, its implementation has a cross-platform functionality that can be run on any operating system. The system is developed in an IoT fashion and the reports can be accessed from the WWW. In this work, the RFID reader is integrated with a microcontroller and multiple readers are connected to the WWW and a database server through Ethernet. The server acquires the attendance data from each reader in a separate thread. The problem with this system is that when the WWW server has crashed all information will be lost accordingly. This is due to the lack of reliable cloud services inside the institution. Moreover, this implementation makes the network busy due to recording the attendance per person for each entry in the database server. Besides, in the case of network crash between the readers and the server, the attendance is stored locally inside the controller during the lecturer time; if the network problem is not solved during the lecturer time, all data will be lost.

Mahmood et al [10] proposed a general-purpose infrastructure for short cut development cycle in RFID's application areas, that is called the infrastructure for RFID-based applications (IRFID). This infrastructure is reused for attendance and student tracking inside the University. In this system, the

RFID readers are attached directly to an edge server which is a personal computer (PC). Like IAAS, IRFID ensures power saving for the RFID readers by turning the readers' power on during the lecturer hours. This system provides real-time reporting and tracking and, on the other hand, it has no absent management which is required further programming models. Like the IAAS, the implementation is based on a local PC to store the database tables instead of the cloud. A PC is used to host the RFID readers and there is no fog computing to handle the database backup operations. In the case of a network failure, all the data will be lost until the networking problem is resolved.

Sharma and Aarthy [11] handled the time-wasting problem of taking student attendance each lecture manually through using technology. An Automatic Attendance Monitoring (AAM) was implemented based on a combination of the benefits of RFID and IoT. Besides, the attendance tables are stored on a cloud computing, which ensures flexibility and gives better proficiency. However, AAM is assumed to store the data directly to the cloud which increases the risk of losing data due to connection loss to the cloud during the time of sending data. Furthermore, the design of AAM is based on attaching the readers to a PC and sending the data to the cloud in a real-time fashion. Like IRFID, the AAM system does not have the absent management which is required further programming models. Furthermore, there is no fog server in this system to handle the database backup operation.

Sunehra and Goud [12] introduced an affordable and user-friendly attendance system built to solve the manual attendance recording problem. Their proposed system is called attendance recording and consolidation system (ARCS). A Raspberry Pi is used as a Web server that obtains attendance data from an Arduino platform which work as an attendance recording device (ARD). However, the ARCS is susceptible to failure since it depends on Raspberry Pi as the main server that must perform the calculations of all the attending students, which is considered to have low computational capability as compared to real servers or even PCs. Like IASS, this system has absent management for real-time reporting. On the other hand, there is no fog computing to handle database backup operations. Besides, the readers are always independent of the operation time. Moreover, this system is subject to fail if the main server is crashed. Finally, In the case of a network failure, all the data will be lost until the networking problem is resolved.

Irawan et al. [13] proposed a real-time attendance system based on RFID as well as IoT and cloud computing. This Attendance monitoring system (AMS) starts with a login page to open the next page depending on the role of the user, i.e. whether a student or professor. The attendance is collected by RFID and stored in a cloud computing during all the periods of time included. Thus, the system does not provide power-saving and database backup. Moreover, the AMS system is subject to losing data when the network is failed, as in the previous systems.

Kavitha and Sankara [14] proposed a Smart Attendance System (SMS) that collects attendance based on the use of RFID and fingerprint attached to a Raspberry Pi. The SMS works on sending an alarm message to those to reach a risk ratio of absence during the lecturer time. Thus, this system has a power saving functionality. However, this system does not scale well due to the absence of cloud reliability. Moreover, this system does not support yearly and monthly reports of attendance, while the database backup operation is also missing. Finally, there is a possibility of losing data when the network is failed, as in the previous systems.

All the valuable works discussed above have a low level of attention to deal with network failure as well as database backup operations, which act as common deficiencies for the reviewed works. As such, this paper contributes to tackling these issues.

AASCF Architectural Design

The AASCF system consists of three main components, namely the Client_side, Server_side, and Fog_server. These Client_sides are connected by Ethernet to both the Fog_server and Server_side, as shown in Figure-2. The following sub-sections show the system operation scenarios for each part.

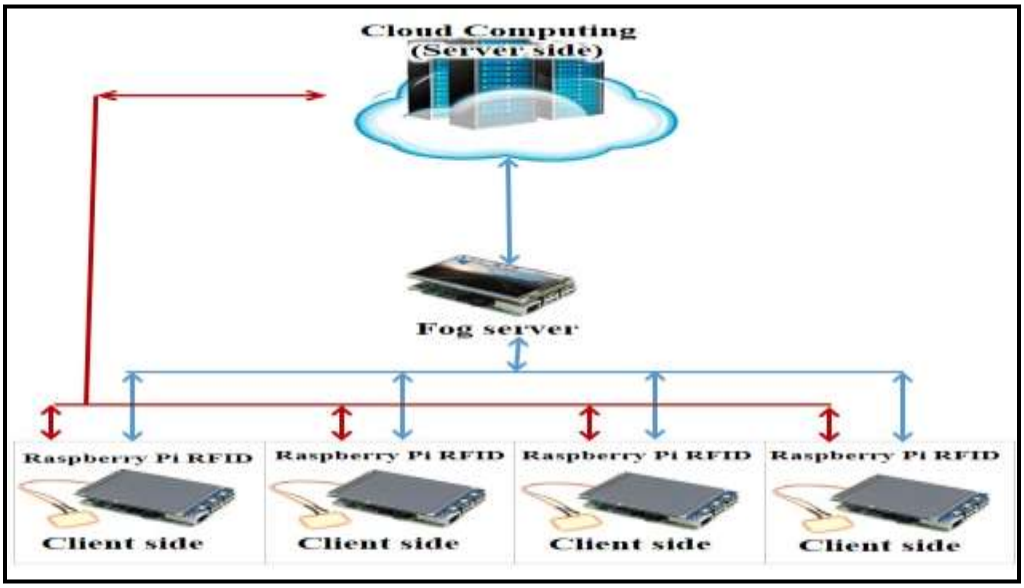

Figure 2- AASCF architectural design.

Client-side

The Client-side consists of an RFID reader with low frequency (RDM630 RFID reader module), RFID antenna, and Serial–USB converter (PL2303HX), hosted by Raspberry Pi. The realization of the hardware circuit interface is shown in Figure-3. The Client-side is installed for each lecture holes in the departments and interacts with the users using touch-screen LCD.

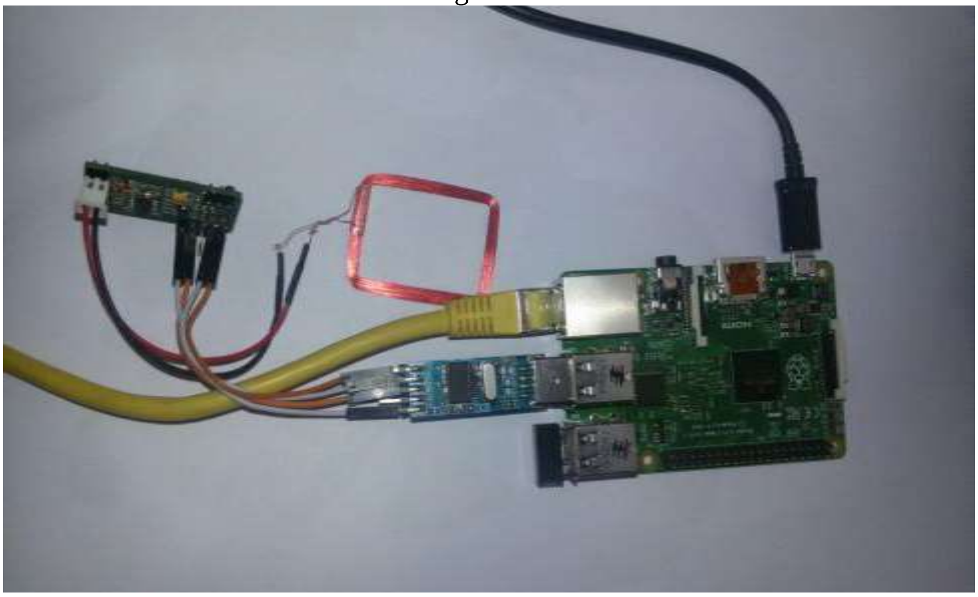

Figure 3- Connection of RFID to Raspberry Pi.

At the start of the official working day at the university, the system is waiting for a professor to start the lecture. The lecturer starts when the lecturer presents his/her tag to the RFID Reader, where the lecturer time will be compared to the current day and time to start the lecture. The confirmation will end when the lecturer confirms the correct time and date. Another window will be displayed containing the start time of the lecture, the end time, and the title of the lecture. Next, the RFID Reader reads the tags of the students and records attendance in the Database, as shown in Figure- 4. At the end of the lecture the professor presses the end button to see the absentee list and to confirm the attendance record properly. If there is a student taking a vacation or there is someone who forgets his/her tag, the professor registers his/her presence by pressing the clearance button.

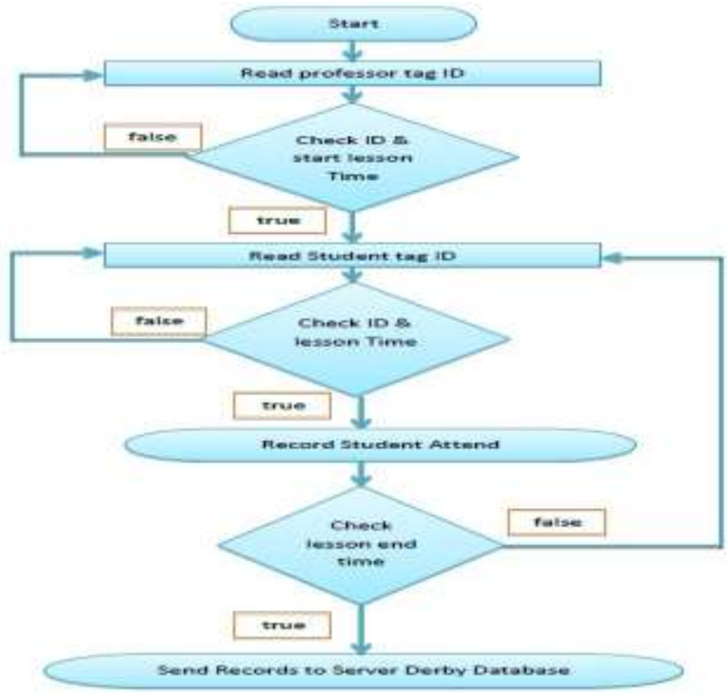

Figure 4- Flowchart for Client-side system operation.

Thus, the client-side operations can be stated as follows.

1- When the lecture starts the professor enters his tag to allow starting the record attendance. Each student should pass his/her tag on the RFID device that is connected to the Raspberry Pi device that works as a client. During this time, the attendance is stored inside the Raspberry Pi local database.

2- At the end of the lecture or when the professor presses the end button, the client will send the records to the Fog-server for storing. If the Fog-server is unreachable, the client will send the records to the Server-side directly for storing and reporting purposes.

Server-side

The user can access the server via a user name and password. The server program consists of several graphical user interfaces. For instance, the department's head or coordinator has the rights to edit the articles, add students, and modify absenteeism rates, the schedule, and the materials. The professor is authorized to enter only the material that he/she teaches. Also, the professor is allowed to edit the absences in the case of a mistake happened. Finally, when the server receives the records and they are stored inside the cloud, the Server-side issues a delete command to delete these records from both Fog_server and Client_side. Thus, Server-side operations can be stated as follows.

1- The head of the department can update the attendance after logging in with the administrator's privilege.

2- Each professor can update the attendance list for the subject by entering the subject title and his/her credentials.

3- Delete commands are issued to delete records from Fog_server and Client_side when the records are stored in the Server_side.

There are three roles with different permissions in this system. The first one is the head of the department that has the permission to insert and update data related to students, daily schedule, and subject, along with updating the attendance for the students in any subject. The second permission is given to the professors who can only view and update the attendance of student of the subject they are responsible for. Finally, the last permission is given to any subject who can only view the attendance without making any updates.

The use case diagram for both the Client_side and the Server_side is illustrated in Figure- 5.

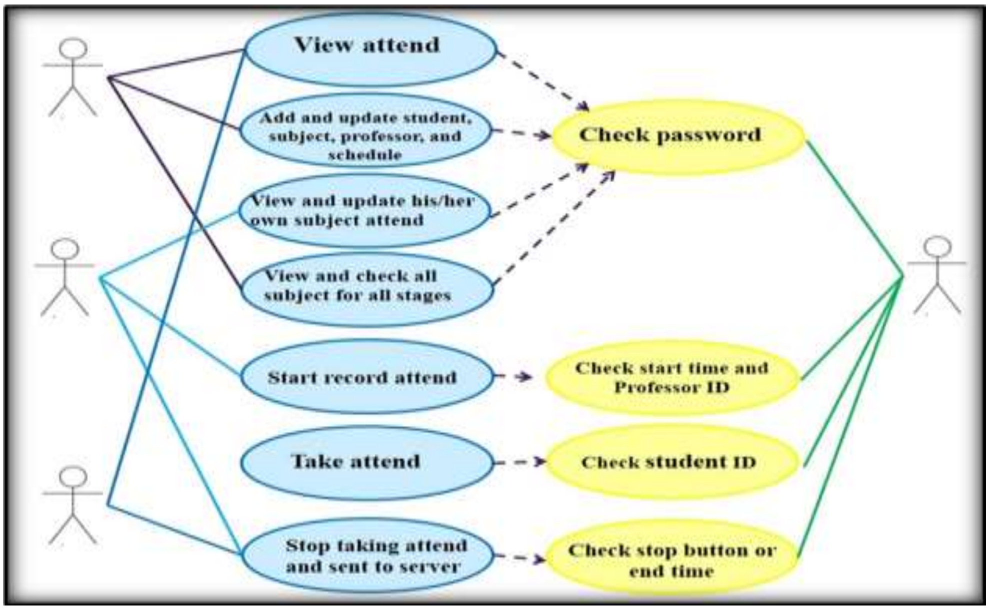

Figure 5-Attendance system's use case diagram.

Fog-server

The Server-side is hosted by the cloud computing while the Client-side is hosted by the Raspberry Pi for each classroom. Between these two sides, there is a Fog-server which is hosted by a Raspberry Pi that works on collecting data from the Client-sides of the classrooms.

At the end of the lecture, the collected records will be sent from the Client-side to the Fog-server, which in turn sends these records to the cloud computing (Server-side) at the end of the official university time. This scenario will have a high level of efficiency in reducing the load on the cloud by using the Fog-server. Besides, it is very useful if one or more of the classrooms' links to the cloud is crashed. In this case, the records will be sent to the Fog-server and will not be lost. Figure- 6 illustrates the sequence diagram for the dataflow in the AASCF.

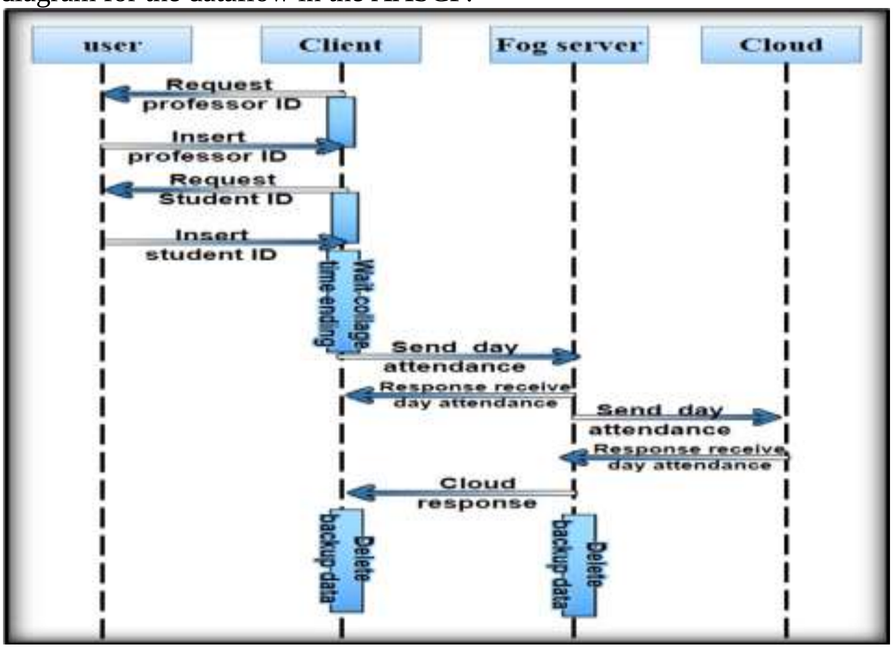

Figure 6- The sequence diagram for the dataflow in the AASCF.

Software Implementation

Graphical user interfaces (GUIs) are needed for client and server applications. While the Fog_server applications have simple command-line interfaces (CLIs) that manage the dataflow of the records and issue the delete commands.

These applications are written and tested using Java programming language and NetBeans IDE. Java is an interpreted language so the code is “written once and could run anywhere (WORA)” with the help of JVM. Furthermore, the Derby database is selected for the realization which is built inside the NetBeans IDE. The GUIs are implemented using JavaFX package.

The GUI which is run in each classroom inside the Raspberry Pi along with the responses to the lecturer’s events on the LCD are shown in Figure- 7. Here, it should be mentioned that the Client-side can be also run on a PC without any change. However, the cost of a PC is significantly higher than that of the Raspberry Pi.

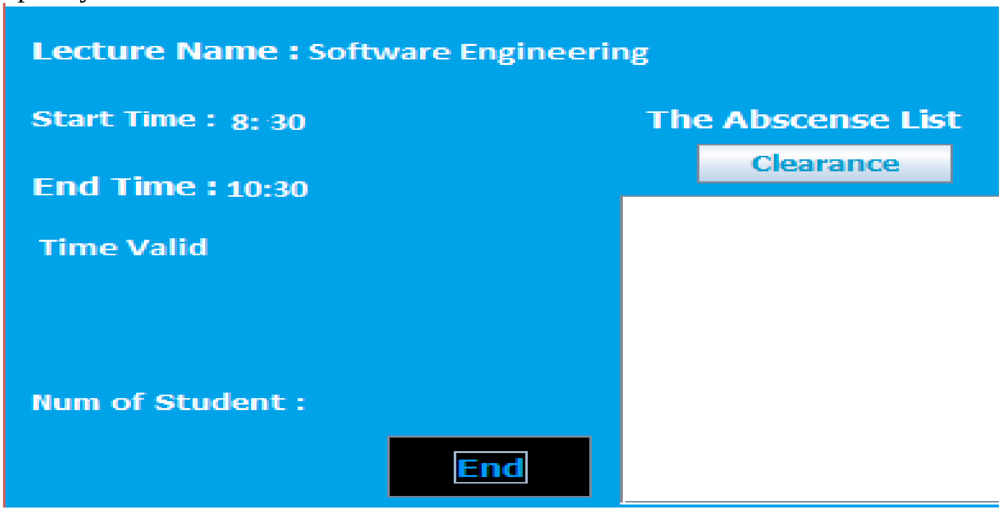

Figure 7- A snapshot for the Client_side window as displayed on the Raspberry Pi’s LCD.

The Server-side is responsible for the calculations and viewing the results to those who are concerned. The attendance application’s Server-side starts with a login page that decides the roles, as shown in Figure-8.

Figure 8- A snapshot for the login page at the Server-side.

The software on the Server-side enables the head of a department to add subjects and control the lecture time. Moreover, the reports for the student attendance in each class are displayed, in addition to the record updates by the professor or by the head of the department, as shown in Figure-9.

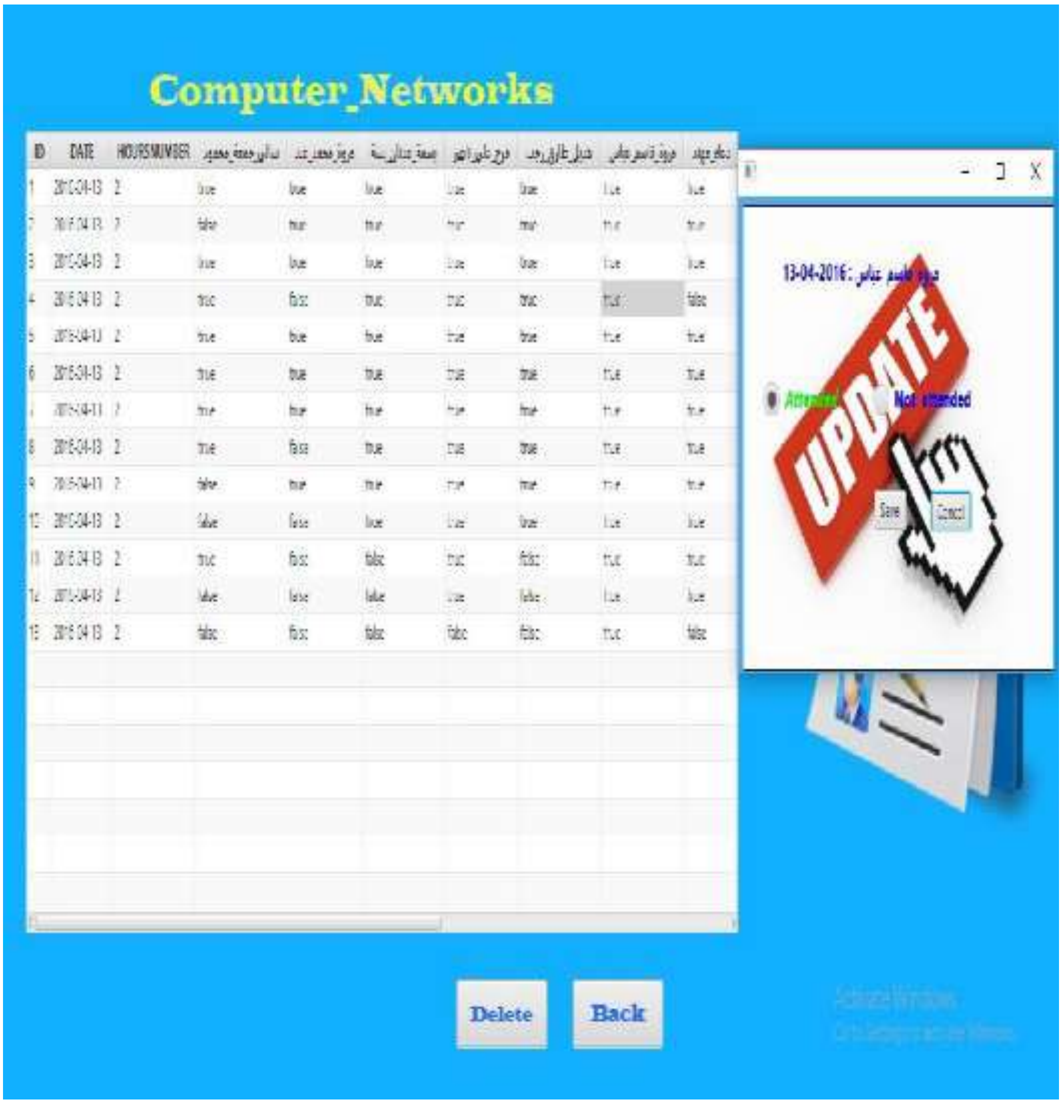

Figure 9- A snapshot for updating the student attendance at the Server-side.

The AASCF Evaluation

In this section, the response of the proposed AASCF in the case of failures is discussed. Besides, a comparison with the related works is conducted.

Response to failures

The following three cases of failures must be kept in mind when evaluating the system as follows. Case 1, the connection is lost between the Client-side and the Fog server. Case 2, the Fog server is crashed. Case 3, the connection to the Server-side is lost. The proposed AASCF provides a solution for each of these three cases, as follows.

Case 1: To avoid the loss of the attendance data, a copy of attendance is kept in the Client-side until the end of the day. The data will be sent to the cloud after it is being sent to the Fog-server. The Client-side waits to get a response from the Fog-server. If this response is coming, the status of data will be pointed as a "sent-to-Fog". But if there is no response came from the Fog-server, the data will be kept as "unsent" status.

In this case, a thread will try to send the records again after one minute in case the connection is back. This trial will be repeated one time after each lecture and at the end of the official university time. The number of trying is increased to five-times for five minutes only (try each one minute). The roles of the Client-side networking are depicted in Figure-10.

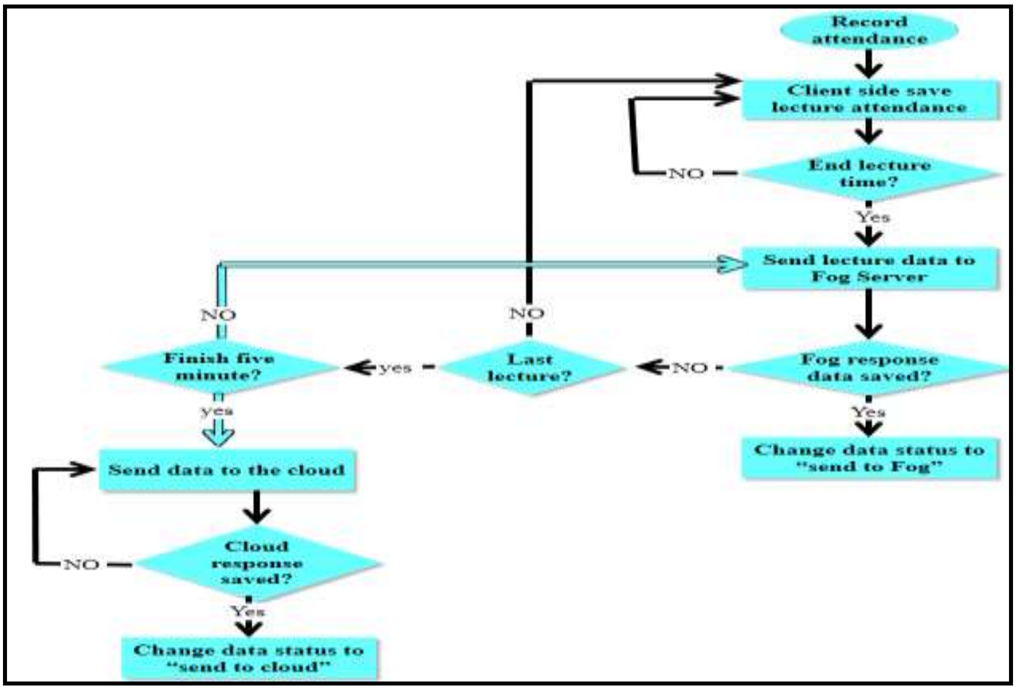

Figure 10-The roles of Client-side networking.

Case 2: If the five minutes are passed with no connection to the Fug-server or the Fug-server is crashed, the status of data on the Client-side will still on "unsent" state. In this case, the client will send the data directly to the cloud and changes their status to "sent-to-cloud" (the red-lines in Figure-2). Now, when the Fog-server's connection is back or the Fog-server recovers to a working state, it checks the status of the clients and decides whether, or not, to send the data to the cloud. If the status was "sent-to-cloud", the Fog-server will delete the data from the Client-side at the end of the official University time or after the recovery event.

Case 3: The data is kept in both senders (Fog-server and the Client-side) until it is sent to the cloud and a response is received that it had been saved in the cloud. If the connection has been lost between the cloud and the senders, the data will not be deleted from these two senders even if the day is over, rather the data will be kept and new tables will be added. When the connection is back, all the data will be sent again as illustrated by the blue-lines in Figure-2. After saving the data into the cloud, the Server-side issues a corresponding command to the Fog-server to delete these tables from the corresponding sender(s).

Evaluation and comparison

As seen in the aforementioned discussion, and as compared with the literature review section, the proposed system AASCF system, unlike the reviewed systems, handles networking connection failures and device crashes systematically by providing different storage levels and multiple servers (Fog and cloud). Moreover, the AASCF system provides a protocol for database backup and storage in a hierarchal form through the intermediate Fog-server. Thus, these features are unique to the AASCF system.

The AASCF system, like AAM and AMS systems, but unlike IAAS, IRFID, ARCS, and SMS systems, stores the database on the cloud.

The AASCF, IAAS, and ARCS systems, unlike IRFID, AAM, AMS, and SMS systems, provide the calculation of student absence ration automatically. All the systems, except for ARCS and AMS, provide the power-saving feature because the hosting devices are powered on depending on lecture time.

All the systems host the RFID reader by a microcontroller, except for IRFID and AAM systems, in which the RFID reader is hosted by a PC. The hosting by a controller reduces the realization cost significantly. Besides, unlike the reviewed system, in the proposed AASCF, both hosting types can be applied without changing any source code; Only Raspberry Pi could be changed to be a PC.

The comparison among these systems is tabulated in Table-1.

Table 1-Comparison between the Proposed AASCF System and Related Works' Systems.

Supported Feature ✓ Not Supported Feature × Can be Supported Feature *	Automatic Attendance Systems						
Systems / Functions	AASCF	IAAS [2]	IRFID [10]	AAM [11]	ARCS [12]	AMS [13]	SMS [14]
1- Cloud base	✓	×	×	✓	×	✓	×
2- Attendance full management							
➢ calculate student absence ratio	✓	✓	×	×	✓	×	×
➢ Power saving	✓	✓	✓	✓	×	×	✓
3- Handle connection loss	✓	×	×	×	×	×	×
4- Database backup	✓	×	×	×	×	×	×
5- Hosting the RFID Reader							
➢ PC	*	×	✓	✓	×	×	×
➢ Microcontroller	✓	✓	×	×	✓	✓	✓
6- Fog-server	✓	×	×	×	×	×	×

Conclusions

Attendance recording is important in most life fields and, as a result, it is important to build a system that can do such a process with a low-cost and flexible manner. Hence, this paper proposed a system that collects student attendance data based on their tags. Also, this paper proposes a novel feature for network failure tolerance to tackle network failure issues. A Fog server is used to reduce the load on cloud computing besides having a copy of attendance data from a Client-side, in a way that ensure that the data reaches the cloud whatever happens to the connection or the Fog-side, or even to the Clients-side. Some feature can be added in the future, such as using physical identification with the RFID tag to prevent imposter. Another future direction is to add the load balance capability inside the cloud computing to handle all the requested calculations. Finally, it is possible to increase the reliability and availability of the system by making a backup of data inside the cloud and making redundant Fog-servers.

References

1. Sari, A. C., Rahayu, A. and Budiharto, W. **2015**. Developing Information System of Attendance and Facebook Status for Binus University's Lecturer Using Raspberry Pi Architecture. Procedia Computer Science, Elsevier, **59**: 178-187. doi: 10.1016/j.procs.2015.07.546.
2. Younis, M. I. , Al-Tameemi. Z. F. A. , Ismail, W. and Zamli, K. Z. **2013**. Design and Implementation of a Scalable RFID-Based Attendance System with an Intelligent Scheduling Technique. *Wireless Personal Communications*, **71**(3): 2161-2179. doi: 10.1007/s11277-012-0929-3.
3. Alshoudokhi, Y. A., Ashraf, M. A., AlShareef, M. R. and Behairy, H. M. **2019**. Recent Developments in Chipless Ultra-Wide-Band (UWB) Radio Frequency Identification (RFID) Systems. Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 535-538. doi:10.1109/IAEAC.2017.8054072.
4. Popat, P. , Sheth, P. and Jain, S. **2019**. Animal/Object Identification Using Deep Learning on Raspberry Pi. *Information and Communication Technology for Intelligent Systems* (ICTIS), **106**: 319-327. doi: 10.1007/978-981-13-1742-2_31.
5. Murthy, M. N. and AjaySaiKiran, P. **2018**. A Smart Office Automation System Using Raspberry Pi (Model-B). International Conference on Current Trends towards Converging Technologies (ICCTCT), 1-5. **doi**:10.1109/ICCTCT.2018.8550894.
6. Kumar, P. and Kumar, R. **2019**. Issues and Challenges of Load Balancing Techniques in Cloud Computing: A Survey. *ACM Computing Surveys* (CSUR), **51**(6): 1-35. doi: 10.1145/3281010.
7. Rashid, A. and Chaturvedi, A. **2019**. Cloud Computing Characteristics and Services: A Brief Review. *International Journal of Computer Sciences and Engineering* (JCSE), **7**(2): 421-426. doi: 10.26438/ijcse/v7i2.421426.

8. Al Yami, M. and Schaefer, D. **2019**. Fog Computing as a Complementary Approach to Cloud Computing. International Conference on Computer and Information Sciences (ICCIS). doi: 10.1109/ICCISci.2019.8716402.

9. Khattak, H. A. , Arshad, H. , Islam, S., Ahmed, G., Jabbar, S. and Sharif, A. M. **2019**. Utilization and Load Balancing in Fog Servers for Health Applications. *EURASIP Journal on Wireless Communications and Networking* (JWCN). 1-12. doi: 10.1186/s13638-019-1395-3.

10. Mahmood, B. M. R., Younis, M. I. and Ali H. M. **2013**. Construction of a General-Purpose Infrastructure for Rfid–based Applications. *Journal of Engineering*, **19**(11): 1425- 1442.

11. Sharma, T. and Aarthy, S.L. **2016**. An Automatic Attendance Monitoring System using RFID and IOT using Cloud. Online International Conference on Green Engineering and Technologies (IC-GET), doi: 10.1109/GET.2016.7916851.

12. Sunehra, D. and Goud, V. S. **2017**. Attendance recording and consolidation system using Arduino and Raspberry Pi. International Conference on Signal Processing, *Communication, Power and Embedded System* (SCOPES), 1240-1245. doi: 10.1109/SCOPES.2016.7955639.

13. Irawan, J. D., Adriantantri, E. and Farid, A. **2018**. RFID and IOT for Attendance Monitoring System. *MATEC Web of Conferences*, **164**: 1-8. doi: 10.1051/matecconf/201816401020.

14. Kavitha, G.and Sankara, K. **2018**. Design and Implementation of Smart Attendance System using Raspberry Pi. International Journal of Scientific Research in Computer Science, *Engineering and Information Technology* (IJSRCSEIT), **3**(4): 26-33.