Journal of Engineering Science and Technology
Vol. 18, No. 1 (2023) 257 - 275

DESIGN OF ATTENDANCE INFORMATION SYSTEM USING RFID

W. C. C. CHOE[1], Y. H. TEH[1], Y. W. LOW[1], T. ELTAIF[2], K. N. MINHAD[1,*], J. D. TAN[1], M. A. S. BHUIYAN[1]

[1]Department of Electrical and Electronics Engineering, School of Electrical Engineering and Artificial Intelligence, Xiamen University Malaysia, Jalan Sunsuria, Bandar Sunsuria 43900 Sepang, Selangor, Malaysia
[2]Faculty of Engineering Technology and Science, Higher Colleges of Technology, Madinat Zayed, Al-Dhafra, Abu Dhabi, United Arab Emirate
*Corresponding Author: khairunnisa.minhad@xmu.edu.my

Abstract

Radio frequency identification (RFID) technology is an eminent wireless system that functions with the help of a reader and a tag. Implementing this technology could lead to developing a sustainable student attendance system. Therefore, this paper vividly captures the fact that the development of student recognition and registration system based on RFID technology and the embedded system can deliver tangible benefits to globalisation and environmental sustainability. The design of the attendance system based on RFID technology verifies the students' identification, venue, time and date upon receiving the transmitted information from the respective RFID tag embedded into the student card. The implemented system includes the embedded board Arduino Mega 2560, RFID reader MFRC522, time-keeper DS3231 Real Time Clock (RTC) Module, 20×4 LCD with I2C, network adapter Arduino Ethernet Shield R3 and 9V AC/DC power adapter 0910. The modern software tools adopted are Arduino Integrated Development Environment (IDE), XAMPP, Apache HTTP Server, MySQL and PHP MyAdmin. The proposed system is developed and tested in the institution, and the users of the system consist of admin staff, lecturers and students. The proposed system provides a robust and automatic attendance system that deals with the problems of the traditional attendance system. The results show that the developed method is practical, efficient, ready-to-deploy, and error-free as a 100% success rate is achieved. The interfacing device can synchronise its time with the webserver and send attendance verification requests successfully without any error.

Keywords: Arduino Mega 2560, Attendance system, Embedded system, Radio frequency identification, System verifications.

1. Introduction

Globalisation and technological advancements are the key factors that drive our current times and affect the future of education learning approaches [1]. There has been a boost in the development of technologies that could project a significant impact on society and the environment in today's digital world [2]. Biometrics approaches such as face recognition, fingerprint, Bluetooth, Near-Field Communication (NFC) and Radio Frequency Identification (RFID) have been adopted to reduce the administrative complexity and cost of the traditional attendance system and increase its efficiency [3, 4]. However, among the many options available in the market, the combination of RFID and microcontroller is one of the approaches that can reduce system cost and human error in an automated recognition and registration management system. The RFID technology is widely used in a diverse array of applications. RFID is a part of the Automatic Identification and Data Capture (AIDC) technologies family, which is proven to be fast, reliable and accurate in object identification [5].

The RFID technology operates in various frequency bands, including the low frequency (LF) band, high frequency (HF) band and ultra-high frequency (UHF) band that covers different frequency ranges [6]. The RFID reader is responsible for transmitting a radio wave pulse to the RFID tag that falls within its coverage and anticipates response [7]. The integrated RFID reader, which has a built-in antenna and an additional antenna port, is the common subset of the fixed readers [8]. The number of extra antennas connected to the reader depends on the coverage area required for the RFID application [9]. The mobile RFID readers are flexible, portable, handheld devices that enable the user to switch locations according to their needs. The mobile readers are categorized into two categories; Mobile Computing devices (connected to an on-board computer) and Sleds (connected wirelessly to a smart device or tablet via Bluetooth or using an Auxiliary connection) [10, 11]. The RFID tag comprises two parts: the antenna and an RFID chip. The antenna receives radio waves signals and transmits signals back to the reader as a response. At the same time, the RFID chip, an integrated circuit, is responsible for storing the identification (ID) of the tag and other relevant information that is needed [12, 13].

Kong et al. designed a student attendance system that uses an RC522 RFID reader, Visual C++ as the development tool and MySQL as the database [14]. Every attendance information received from the reader will be saved and stored in the real-time database without human interference. Sharma et al. proposed an automated attendance monitoring system using the cloud using RFID and the Internet of Things (IoT) [15]. This operation will enable the authorized users to access the data anytime, anywhere to reduce the cost of consumption compared to the use of computers in maintenance and the purchasing price. The RFID technology also has been used based on employee attendance system using VB.Net language [16], CSS, HTML and PHP language [17], Python and MySQLite3 [18], C# and VB.Net incorporated with Microsoft Access database [19]. There are various applications, including developing an intelligent attendance system based on a frequency distribution algorithm with passive RFID tags [20], a smart mobile interfaced RFID-based attendance system [21], an IoT-based smart attendance system using RFID [22], RFID attendance system with face detection using validation Viola-Jones and local binary pattern histogram method [23], MQTT based IoT-RFID attendance system [24], telemonitoring attendance system using

RFID and Photo-Cell [25] and RFID based attendance system using temperature and sanitiser [26]. As more attention has been brought to the development of RFID, this device has tremendously improved processing speed, memory capacities and reading range. Therefore, this paper proposes a design of students' attendance system based on RFID technology that verifies the students' identification, venue, time and date upon receiving the transmitted information from the respective RFID tag embedded into the student card.

This paper highlights the importance of verifying the identification of the attendee. Referring to the related works presented, many researchers focused on the attendance system and neglected the check-in verification process as they only concentrated on accepting the information transmitted to the reader. The method of verifying if an attendance request from the interfacing device is valid at the server-side begins with the four essential parameters sent along with the request. Student identification (ID) number, venue, date, and time are needed to check if a class is going on in the classroom. When the server has received the request with these four parameters, it will first check if there is a match found in the table venue_detail, then retrieve the corresponding ongoing lecture's course code if there is a match, else echo "0" to the client (interfacing device). The last important parameter is the Student ID which is needed to ensure that the student has taken the subject. The server will check for it by searching the table subjects_student. If all the criteria are met, the system will mark the student's attendance and echo "1" to the client. Else echo "0".

2. Design Framework and Implementation

2.1. System architecture

This recognition and registration system's development and design comprise hardware and software aspects. The hardware uses six components, including the embedded board Arduino Mega 2560, RFID reader MFRC522, time-keeper DS3231 Real Time Clock (RTC) Module, 20×4 LCD with I2C, network adapter Arduino Ethernet Shield R3, and 9V AC/DC power adapter 0910. Arduino Mega has been selected because of its larger RAM and flash memory capacity and has more digital pins than other Arduino boards. The RFID reader MFRC522 operates at 13.56 MHz and supports ISO/IEC 14443 communication standard. These two crucial specifications determine what type of RFID tag will be used in this system.

The time-keeper DS3231 Real Time Clock (RTC) Module is preferred as a solution for time-keeping as a high-frequency oscillator is needed to keep time accurate. The module for DS3231 RTC has a local EEPROM, power supply and battery, which means that the time is still running even when the Arduino Mega is power-off. Moreover, a simple LCD with a matrix display of 4 rows and 20 columns is used to display time and other necessary information for its user. The Arduino Ethernet Shield R3 is selected as the network adapter for communication with the webserver. It has an RJ45 port connecting with an Ethernet cable and an SD card reader. The AC/DC power adapter 0910 has been selected as a power source for the Arduino board. On the other hand, in the aspect of the software, the Arduino Integrated Development Environment (IDE) is used for writing Arduino code. It is a cross-platform application written in functions from C and C++ programming languages. The Arduino IDE is applied to write and upload programs to the Arduino Mega 2560. Besides, the free and open-source cross-platform web server software was used to design the web server, XAMPP, Apache HTTP Server, MySQL and PHPMyAdmin.

Figure 1 shows the block diagram of the overall system architecture. The webserver stores essential data such as students' attendance, classroom schedules, time, and other necessary information for the check-in verification process. The verification process is a task that the webserver will carry out to verify if an attendance request sent from the interfacing device is valid. The framework of the webserver can be divided into backend and frontend. The backend framework deals with the database and server-side software.

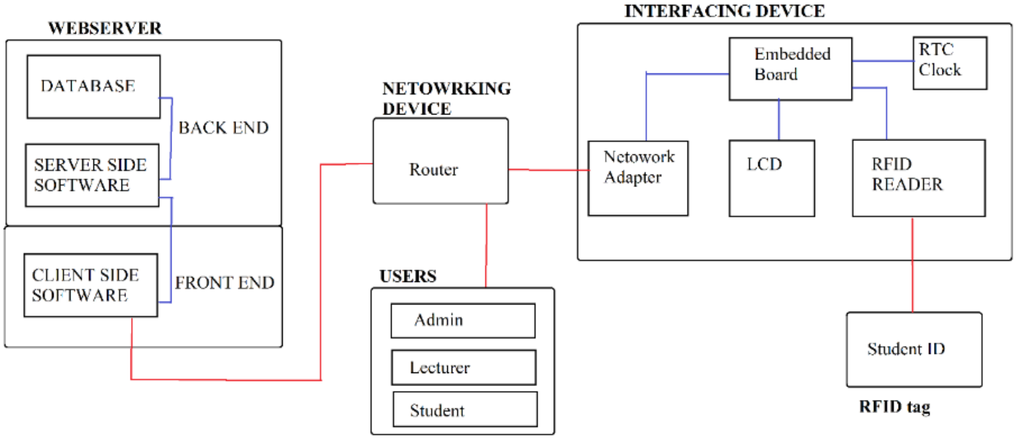


Fig. 1. System architecture. Red lines indicate external communication. Blue lines indicate internal communication.

The database is responsible for storing the critical data, while the server-side software deals with the backend process from the frontend request. The frontend framework is a web-based application that provides a graphical user interface to interact with the system. At the same time, the client-side software is used to build the frontend framework.

The interfacing device is responsible for displaying date and time, reading student ID cards, and sending requests to the webserver for verification or time synchronization. The interfacing device consists of four peripherals connected to an embedded board. The embedded board is a controller that will instruct the peripherals on what to do. RFID tag stores the information of students' ID for the check-in process. The external modules adopted in developing the interfacing device are a real-time clock module (RTC), RFID reader, network adapter, and liquid-crystal display (LCD). The function of the RTC is to store the date and time accurately, while the RFID reader functions to read student ID cards. The network adapter acts as a network connection interface, while the LCD displays the date, time, and system message. The interfacing device does not require any graphical user interface, so it does not need to run any web browser. All it needs is the Ethernet shield that can send HTTP requests to the webserver and receive information from it.

The networking device acts like an agent connecting the interfacing device and users to the webserver. The router's job here is to connect the webserver, interfacing device, and users to the network and route the packets to the proper destination. Figure 2 shows a simple diagram of the check-in process. When a student scans the ID card on the interfacing device, the interfacing device will retrieve the information on the card and send an attendance request and the information to the server through the router. Then, the webserver will process the attendance request by comparing and searching data in the database to determine if the request is valid.

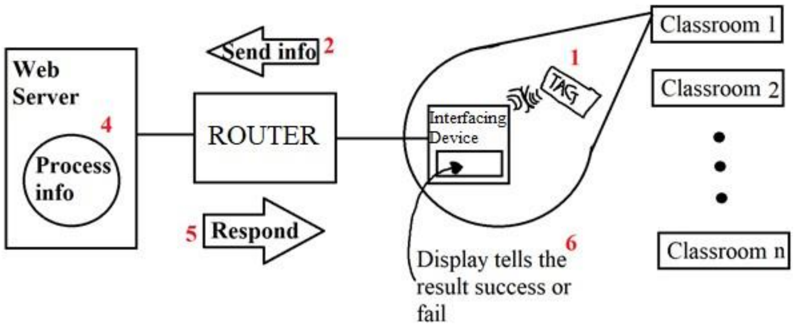


Fig. 2. Check-in process.

After the verification process, the webserver echoes a response to the interfacing device when attendance has been marked. If the attendance request is valid, the student is notified of the class at the right place and time. The only way to set the time on the interfacing device's internal clock module is by synchronizing it with the webserver. The synchronization is done on every start-up of the interfacing device. Theoretically, the interfacing device and web server time should be the same, but time drift is always on both sides. Thus, the verification process uses the date and time on the interfacing device instead of the webserver time. The verification ensures fairness to the students as the webserver time may be a few seconds off from the interfacing device time.

2.2. Webserver

The webserver's framework comprises backend and frontend, further divided into two parts. The first part is the database, while the second part is the server-side software. On the other hand, the frontend framework only consists of client-side software. XAMPP has been chosen to run the webserver as it is free, open-source, easy-to-use and has every tool and software needed. Apache is an HTTP web server software included in the XAMPP package. Apache establishes a connection between a server and a client, while HTTP is the protocol used to communicate between the server and the client.

The database is constructed with MySQL and managed by PHPMyAdmin. The database stores the users' credentials, student attendance, and other necessary information to verify attendance requests received from the interfacing device. The server-side software is a collection of PHP scripts that will handle the requests from the client-side. The scripts are all processed on the server, while the client-side software is responsible for taking the client request and formatting the webpage display layout. The client-side software is written entirely in HTML. However, the client-side software could also add CSS and Java scripts. Figure 3 illustrates the webserver's structure and the tools and languages for each part.

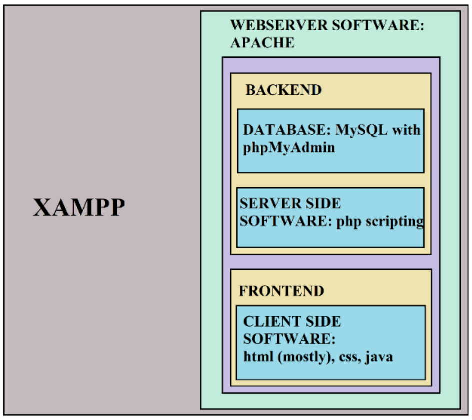


Fig. 3. Webserver structure overview and its corresponding tools and languages.

The login page is developed using an HTML script. When the user submits the ID and password to the web server, the corresponding PHP scripts are executed to check the correct credentials. The PHP scripts will go to the database and search for a match. Assume that the user has submitted a valid ID and password. The PHP scripts will tell the server to direct the user's page to the user's home page. Most of the files are a mixture of HTML and PHP scripts. The server processed PHP scripts, and the client-side processed the HTML scripts. Figure 4 illustrates the process of the system logging in.

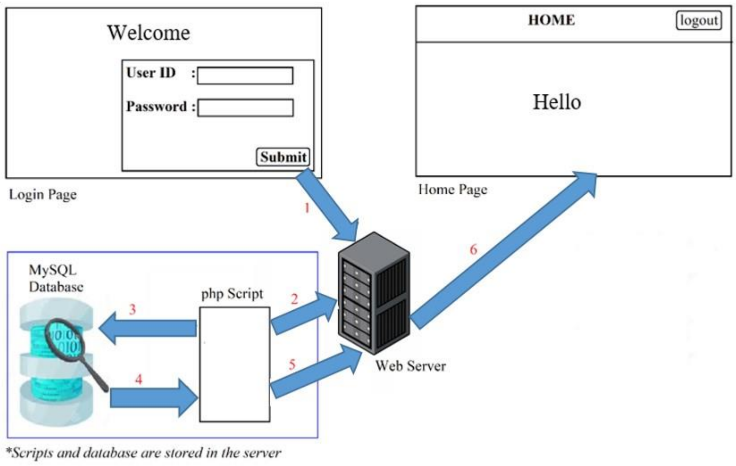


Fig. 4. Simple log in process.

2.3. Webserver users' system

There are two user levels: database-level user and end-level user. The database-level user is directly connected to the database. The database can configure the type of user. For example, a user at the database level can configure to add data to tables, edit the attendance, add on the number of students or classes and view the attendance record. The user at the database level is further categorized into two types of users: admin and regular users. Admin is granted full privileges to the database, whereas normal users can only view what is inside the database. Developers can create, manage, and edit the users' profiles at this level through PHPMyAdmin. The other group of users is the end-level user. This level of users has no direct access to the database. To access the database from the end-user level, developers must connect end-user level accounts to the database level. There are four different types of end-level users shown in Fig. 5.

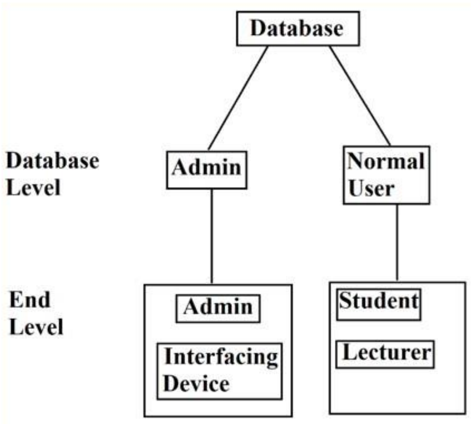


Fig. 5. Types of user at database level and the end level.

Admin is the only person granted the authority to add classes, alter and view the attendance record, and view courses and timetables. The lecturers and students can only view course names, timetables, and attendance records.

2.4. Frontend and backend framework

The design and construction of the backend framework will begin with the database. There are a few parameters to be fixed before further details about the database. The selected parameters are 30 students, ten lecturers, four administrators, six classrooms, and six subjects. Specifying these parameters simplifies the system's complexity yet achieves what is needed for the design of this system. The server-side software and frontend framework contain many files containing HTML scripts, PHP scripts, or both. The frontend framework deals with how the users interact with the system. The design of the frontend framework is about translating the designed web pages into codes. The website's home page has three selections for login, and each section will direct the user to the respective login page.

The webpage redirects to the admin home page with three selections when the admin has successfully logged into the account. The first selection is to view all the available subjects, and clicking on the first selection directs the admin to the subject page. For each subject, the admin can view the subject's registered student, timetable, and attendance. In fact, on the timetable page, the admin can add or drop classes. The second selection on the admin home page shows the student's and lecturer's credentials. Clicking the second selection directs the admin to the student and lecturer page. On this page, the admin can select to view all student's and lecturers' credentials. When students have successfully logged into their accounts, the webpage redirects to their home page with two selections. Students may view their timetable and attendance of enrolled subjects based on the selection. On the other hand, when lecturers have successfully logged into their accounts, the webpage redirects to their home page, which has two choices. The lecturer may then view the timetable and attendance of enrolled subjects based on the selection.

2.5. Verifying attendance requests

The process of verifying if an attendance request from the interfacing device is valid at the server-side begins with the four essential parameters sent along with the request; venue, date, time, and student ID. Venue, date, and time are needed to check if a class is going on in the classroom. When the server receives the request with these four parameters, it will check if a match is found in the table venue_detail and then retrieve the corresponding ongoing lecture's course code. Suppose there is a match, echo "0" to the client (interfacing device). The last important parameter is the Student ID needed to ensure that the student has taken the subject. The server will check for it by searching the table subjects_student. If all the criteria are met, the system will mark the student's attendance and echo "1" to the client. Else quote "0" where "1" represents the success in recording the respective student's attendance. In contrast, "0" means failing to record the individual student's attendance. The flowchart in Fig. 6 illustrates the overall process of verifying students' attendance requests.

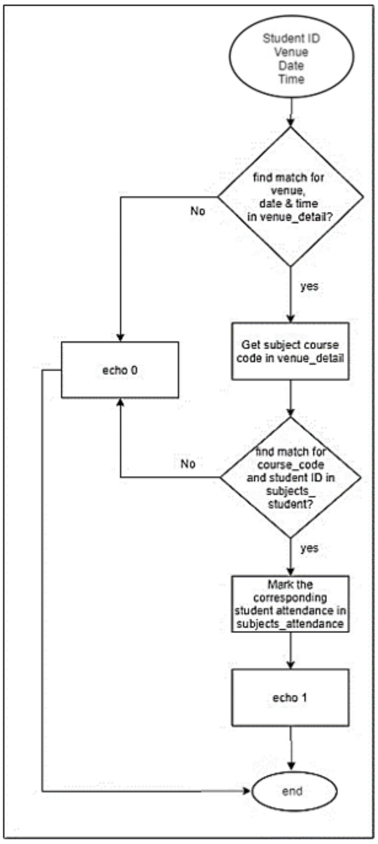
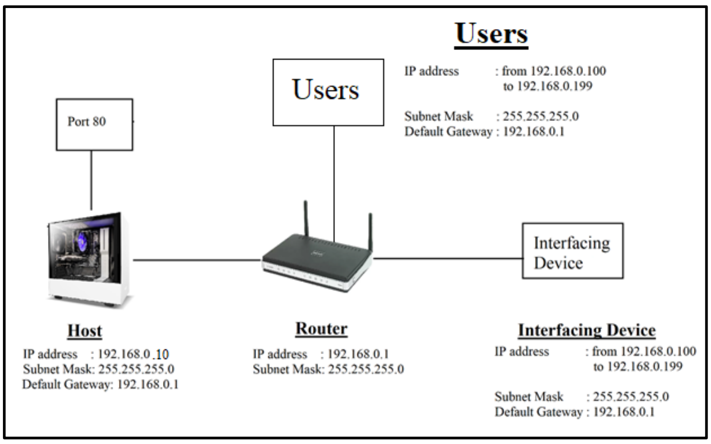


Fig. 6. Flowchart of verifying student's attendance request.

In addition, the process can add features of checking the request. The system recognizes if a student is late to class. For instance, the institution may enforce the regulations on not allowing students to be late to class for more than 30 minutes. The system should allow students to mark their attendance if they are early to the class. If the procedure only allows check-in on time, it would be an issue for a class with many students because it is vital to take note of students queuing up to scan their respective student cards on the interfacing device. Thus, the system should allow early check-in for students to mark their attendance. In the development of this system, there is a time tolerance that enables students to check in 10 minutes earlier.

2.6. System networking

The network applied to this system is the Local Area Network (LAN) which grants access within the selected network. There are a few ways to connect the interfacing device to the webserver. The development of this system focuses on a small scale by implementing two interfacing devices per classroom, with an assumption of 30 students per session. There is an option to select either a router or switch to connect the interfacing device to the webserver. In this work, the router is in this system because it has wireless connectivity. The primary network protocol used in the system is called Hypertext Transfer Protocol (HTTP). In this system, the interfacing device uses HTTP to communicate with the web server by sending an HTTP request. The interfacing device has two different HTTP requests sent to the webserver. One of the HTTP requests is used to synchronize the webserver date and time with the interfacing device, while the other request is used for check-in purposes. Figure 7 shows the network setup and configuration of the system using the format of IPv4 class C. The default gateway address is not essential for the system because this network is LAN and should not communicate with the external network. In subnetting, a network can be subdivided into multiple logical networks, but prior knowledge of which Internet protocol (IP) addresses is required. The IP address is a unique identification address on an IP network, and the address is 32 bits long and is separated into four parts by a period. There are two versions of IP addresses which are IPv4 and IPv6; however, this attendance system only involves the IPv4 address.



Octet	1								2								3								4							
bit	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
value	1	1	0	21 bits																					8 bits (node ID)							

←Network ID→

Fig. 7. Network setup, configuration of the system, and IPv4 class C format.

Each part is called an octet with a decimal value from 0 to 255. An example of a specific IP address assigned to a device in a home network is represented as 192.168.0.157. There are five IP address classes, but only one class is concerned in this system, class C. The first three bits are fixed to 110, which means the first IP address must be 192.0.0.0. The subsequent 21 bits are used to identify the network ID, while the last 8 bits are used for the node. For IPv4 class C, the subnet mask is 255.255.255.0. Applying the subnet mask on the IP address is essentially an "AND" operation. The results are compared using the subnet mask on the IP addresses to determine if the IP addresses are on the same network. Identical results mean they are on the same network, and different results suggest they are on another logical network.

The router's IP address is recorded as 192.168.0.1 by default and directly applied in the system. The subnet mask for this LAN is 255.255.255.0, meaning there can only be 255 devices at maximum within this network. Having a static IP address for the host is crucial because the interfacing device uses it to reference the destination address when sending requests to the server. If the IP address is changed over time, the interfacing device would not know the new IP address assigned to the host. Thus, the interfacing device cannot send any request to the server. The host's IP address is 192.168.0.10, while port 80 is the port number for the HTTP communication between the web server and interfacing device. The IP address of the interfacing device is assigned manually, which is done during the setup phase of the interfacing device.

Manual assignment of the IP address to the interfacing device means that the interfacing device will have a dedicated IP address. The router should reserve the IP address for the interfacing device to prevent IP collision. The Media Access Control (MAC) address is the other important aspect that must be assigned to the interfacing device. The MAC unique address is used as the device within the LAN. The interfacing device uses Arduino Ethernet Shield R3 as the network adapter and has no MAC address. The MAC address format is 00:00:00:00:00:00 in hexadecimal. Users' devices' MAC address is not a concern here since the devices should have their own MAC address assigned by the devices' manufacturers.

2.7. Interfacing devices

The apparatus and components that have been adopted in the development of this attendance system are listed in Table 1. The components used for the interfacing device to develop this system and the circuit diagram for the attendance system are presented. Besides, the setup, looping stage, system flowchart of the interfacing device, and coding for the embedded board are also illustrated. The interfacing device is responsible for reading student ID cards, storing dates and times accurately, displaying necessary information on a Liquid Crystal Display (LCD), and communicating with the webserver. This device is essentially an embedded system.

Arduino board has two stages; the first stage is a setup stage where the Arduino will only run the tasks once and for all, while the second stage is a forever-loop. In this loop, the Arduino will execute the tasks repetitively. Figures 8 and 9 show the system's flowchart. Multiple microcontroller technologies, such as the Raspberry Pi, are implemented in various applications, including mobile robots for surveillance [27]. However, this paper employed the Arduino Mega 2560 as it is sufficient to fulfil this research's purpose due to its larger capacity in both RAM and flash memory and adequate digital pins.

Table 1. List of the main components that are used for the interfacing device.

Device/Component	Name/Model	Functionality
Embedded Board	Arduino Mega 2560 (Compatible)	Controls the peripherals and make decisions
RFID Reader	MFRC522	Reads student ID cards
Time-Keeper	DS3231 Real Time Clock (RTC) Module (included with battery)	Stores date and time
LCD Display	20x4 LCD Display with I2C	Displays information
Network Adapter	Arduino Ethernet Shield R3 (Compatible)	For network communication
9V AC/DC Power Adapter	0910	Supplies DC voltage to the Arduino board

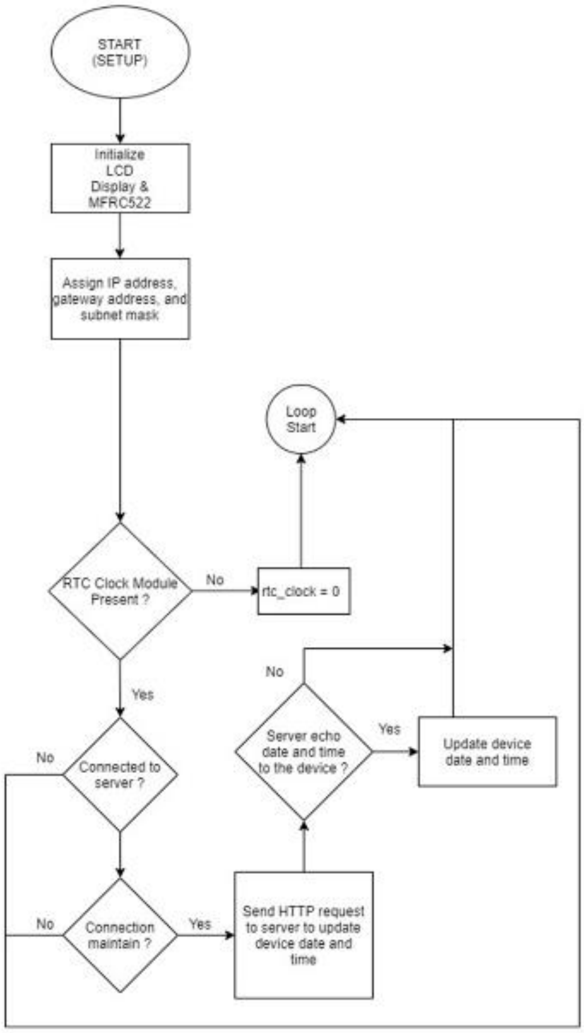


Fig. 8. Flowchart of the system setup.

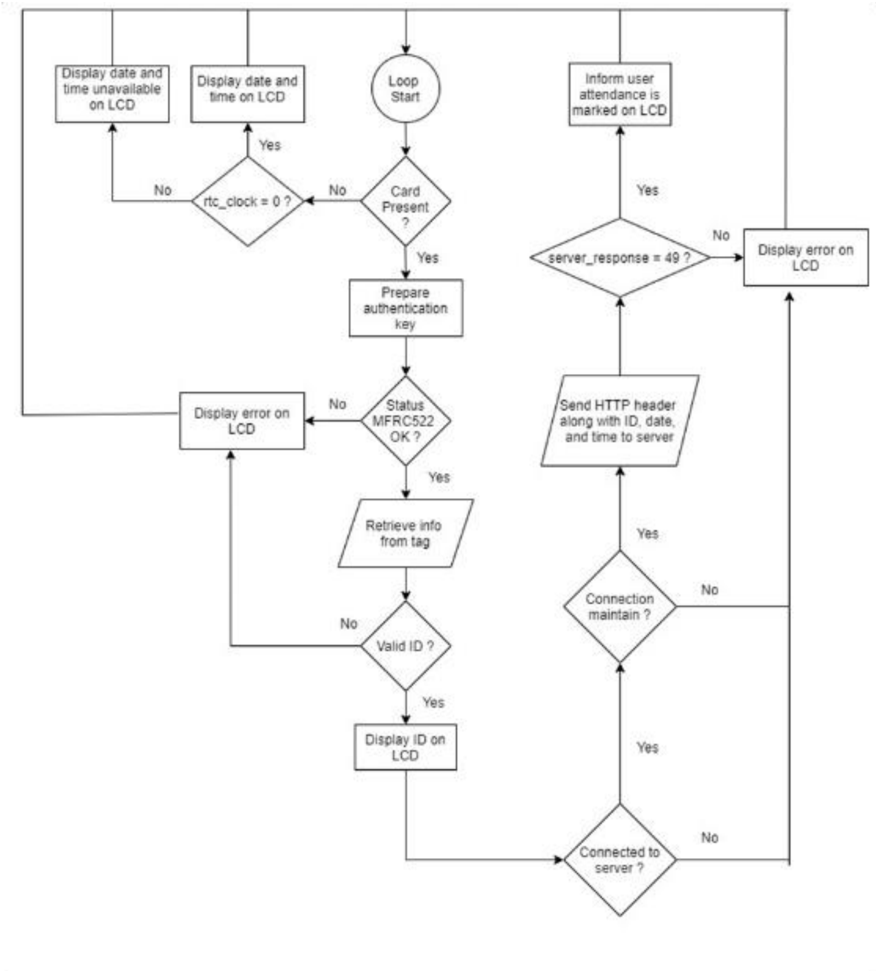


Fig. 9. Looping flowchart of the system setup

2.8. RFID Tag

Mifare Classic 1K has been chosen as the student RFID tag for this attendance system. The 1024 bytes of memory are divided into 16 sectors whereby each has four blocks, and each block is 16 bytes long. The three first blocks of every sector are used for storing data except for the first sector. The first block in the first sector has 4 bytes reserved for a non-unique identifier (NUID), and the rest of the bytes are data written by the manufacturer. The data blocks can be configured as either read/write or value blocks. The read/write block provides storing and reading functions, meaning the user may write or read any bytes on that block. Value block is relatively more complicated and is used for electronic purse functions, error detection and correction, and backup management. The user may define each block and trailer for reading or writing purposes and if key A or B is needed for authentication.

Both key A and B are set to FFFFFFFFFFFF (12 “F”) in hexadecimal at chip delivery. The values may change to some desired values. Key B is unique; if it is not used, it may apply for storing data. Only one piece of information needs to be written on the student ID card, which is the student ID. Note that the student ID here is not a unique identifier for the card but for the students. The total length of the student ID is ten. The first three characters are uppercase letters; the rest are digits ranging from 0 to 9. The ten characters of the student ID are stored in the tag's first ten bytes of the first block in the zeroth sector. The configuration of data blocks, access conditions, and authentication keys are based on the default settings. The data block where the student ID is located is a type of read/write. The data block's access condition where student ID is found read and write if either key A or B is authenticated. Key A remains unchanged at FFFFFFFFFFFF (12 "F") in hexadecimal. Figure 10 is an example of a tag with student ID "EEE1804286".

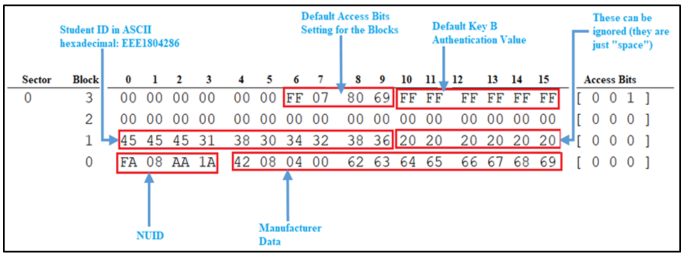


Fig. 10. Result of the data inside sector 0 of a written tag.

3. Results and Discussion

This section presents the performance evaluation of the proposed system from the aspect of the interfacing device and the webserver. The system's performance is evaluated based on the results shown on the web pages and displayed on the LCD of the interfacing device and the data being sent through the serial monitor. Five volunteers from the institution participated in the test. Three students' information was registered into the attendance system database in interfacing device evaluation. All three students are responsible for attending their respective classes based on the given timetable that emphasize the specific classroom (venue), time, date and subject. Compared to the Arduino Uno in the aspect of storage space, dynamic memory and digital input/output pins, an issue may arise related to the bottleneck of dynamic memory. Referring to the Arduino IDE software, the compilation of code on the Arduino Mega has only utilised 10% of program storage space and 21% of dynamic memory, while the compilation of code on the Arduino Uno has used 79% of program storage space and 84% of dynamic memory. It is important to note that the proposed system is ready to be deployed and adopted as a feasible attendance system in the university. The RAM of the Arduino Mega is less concerned as the codes are stored in the server rather than the microcontroller itself. The employment of multiple slaves (receiver) applies in multiple classrooms with only one master. Figure 11 shows the messages printed on LCD during the setup phase. Figure 11(a) shows the successful synchronisation using the server. Figure 11(b) illustrates the date and time on the LCD, at which during this stage, the interfacing device is in the looping phase when a card is tapped onto the interfacing device and the connection to the server is established. The student identification number and device status are displayed on the LCD of the interfacing device when a student taps a valid card.

(a) Successful initialisation

(b) Time synchronisation message

Fig. 11. LCD of the interfacing device.

Figure 12(a) illustrates the date and time on the LCD. During this stage, the interfacing device is in the looping phase, where no card is detected. Figure 12(b) shows the messages printed on the LCD when a valid card is tapped onto the interfacing device and the connection to the server is established. The respective student identification number and device status are displayed on the LCD. Figure 13(a) shows the message printed on the LCD when the attendance of the respective student has been successfully recorded. In contrast, Fig. 13(b) shows a failure case when the student attended the wrong subject or the wrong classroom (venue).

(a) Date and time.

(b) Student ID and device status.

Fig. 12. LCD of the interfacing device.

(a) Successful attendance record.

(b) Failure attendance record.

Fig. 13. LCD of the interfacing device.

Figure 14 shows the serial monitor information sent from the interfacing device during the setup phase. The entire system was tested 35 times repetitively by tapping the tag onto the reader and creating and deleting multiple accounts. The testing phase showed no errors proves that the proposed attendance system has a 100% accuracy and success rate in recording the attendance of the students and the development of the webserver. Figure 15(a) shows the messages sent from the interfacing device to the serial monitor when the attendance is successfully recorded, whereas Fig. 15(b) shows the unsuccessful attendance record.

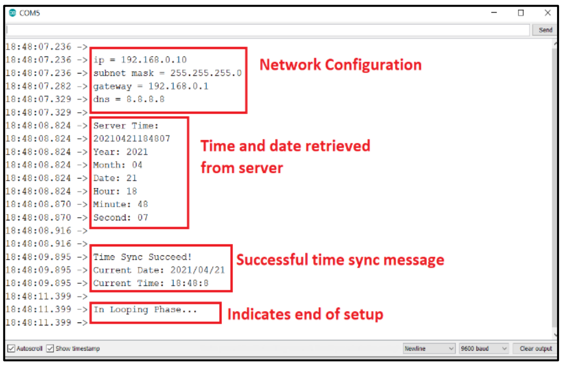
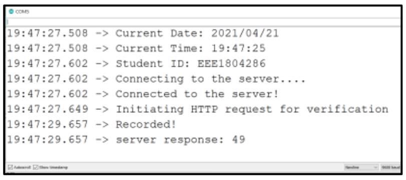
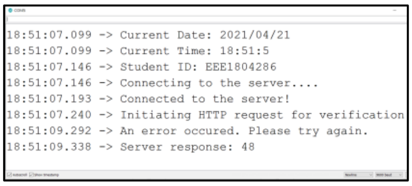


Fig. 14. Messages sent from interfacing device during the setup phase.



(a) Successful attendance record.



(b) Failure attendance record.

Fig. 15. Messages sent from interfacing device.

Timing analysis

The system's processing time to mark a student's attendance is crucial to pay attention to, as an extended processing time may end up in a long queue for scanning students' cards on the interfacing device. There will be some delays in each stage of checking in. There are two types of delays in the system; processing delay and programmable delay. Each processing delay might not be significant enough to be realized, but those tiny seconds that add up might become significant.

Processing delay is inherent in the system. Programmable delay is controlled by the super users in the Arduino coding. The delay () function in Arduino coding tells the Arduino board not to do anything for how long. There are two reasons why such a delay is needed. The first reason is that the interfacing device has an LCD to display necessary information to the students.

The LCD has a minimal size of characters to be displayed. At idle, the LCD shows the date and time, which would take up most of the space on the LCD. The other reason to delay the Arduino coding is to wait for the server's response.

A request sent to the server will need some time to travel along with the network to the webserver, and processing at the webserver takes time. Thus, the response from the server is not immediate and needs a delay in the coding.

Based on the flowchart in Fig. 9, a successful check-in would have two necessary different information displayed and one request to the webserver. Set the delay for waiting for the server's response as 0.5 s and assume the overall processing delay is 1 s at maximum. Then, the total programmable delay for successful check-in progress is based on Eq. (1):

$$\text{Time for attendance marking} = (\text{Time of each information on LCD} \times \text{Number of information}) + (\text{Delay set for waiting server's respond}) + (\text{Other processing delay}) \quad (1)$$

This makes the time taken for the developed attendance marking:

$$\text{Time for attendance marking} = (1 \times 2) + 0.5 + 1 = 3.5 \text{ s}$$

Based on these assumptions, one student would take 3.5s to complete checking in on the system, given that there is no error. Imagine that n students are lining up to scan their student cards on the interfacing device for check-in. When the first one is done, the entire line moves forward and takes one second. Then, the total time for these n students to mark their attendance is based on Eq. (2).

$$\text{Total time for } n \text{ students} = (\text{Time each attendance marking} \times n) + (n - 1) \quad (2)$$

This makes the total time taken for n number of students of the developed attendance marking:

Total time n students attendance marking=4.5n – 1 s

4. Conclusion

This paper proposed developing a student attendance system based on RFID technology. Students can mark their attendance by tapping their ID card embedded with an RFID tag onto the RFID scanner. When students tap their ID card, the webserver will verify certain information to ensure they are in the right classroom and right time for the class. The main contributions proposed to fill the research gap is the verification of time, date, venue, and identification. The development of this attendance system focuses on three significant aspects, including the web server, network connection, and interfacing device. The web server stores data in a database and is responsible for processing the HTTP request from the interfacing device. Besides, the webserver is applied to host a website that allows users to access the database. The network connection connects users and interfacing devices to the webserver. The interfacing device retrieves information from the student ID card, sends it to the webserver, keeps real-time data locally, and displays information on the LCD screen. The interfacing device synchronises its time with the webserver and sends attendance verification requests successfully. This work proves that the proposed approach is practical, ready-to-deploy, and efficient, as no error occurred when experimenting with 30 students attending a lecture. The RFID-based student attendance system fulfils the idea of the "Green Campus" concept to reduce wastage, breed a greener environment, and simplify usage. A sustainable attendance system that uses the current resources available in the university is developed to promote the act of reusing and reducing.

Acknowledgement

The authors acknowledge the support of the grant XMUMRF/2020-C6/IECE/0017 and XMUMRF/2021-C7/IECE/0020 from Xiamen University Malaysia. Funders were not involved in the conduct of the research.

Nomenclatures

n	Number of students

Abbreviations

AC	Alternating Current
AIDC	Automatic Identification and Data Capture
CSS	Cascading Style Sheets
DC	Direct Current
EEPROM	electrically erasable programmable read-only memory
HF	high frequency
HTML	Hypertext Markup Language
HTTP	HyperText Transfer Protocol
ID	Identification
IDE	Integrated Development Environment
IEC	International Electrotechnical Commission
IoT	Internet of Things
IP	Internet protocol
ISO	International Organization for Standardization
LCD	Liquid Crystal Display
LF	low frequency
MAC	Media Access Control
MHz	mega hertz
MySQL	My Structured Query Language
NFC	Near-Field Communication
NUID	non-unique identifier
PHP	Hypertext Pre-processor
RFID	Radio Frequency Identification
RTC	Real Time Clock
SD	Secure Digital
UHF	ultra-high frequency
VB.Net	Visual Basic .Net
XAMPP	cross-platform, Apache, MySQL, PHP and Perl

References

1. Salvia, A.L.; Filho, W.L.; Brandli, L.L.; and Griebeler, J.S. (2019). Assessing research trends related to sustainable development goals: Local and global issues. *Journal of Cleaner Production*, 208, 841-849.
2. Filho, L.W.; Shiel, C.; Paço, A.; Mifsud, M.; Ávila, L.V.; Brandli, L.L.; Molthan-Hill, P.; Pace, P.; Azeiteiro, U.M.; Vargas, V.R.; and Caeiro, S. (2019). Sustainable development goals and sustainability teaching at

universities: Falling behind or getting ahead of the pack? *Journal of Cleaner Production*, 232, 285-294.

3. Okokpujie, I.P.; Fayomi, O.S.I.; Ogbonnaya, S.K.; and Fayomi, G.U. (2019). The wide margin between the academic and researcher in a new age university for sustainable development. *Energy Procedia*, 157, 862-870.
4. Hoo, S.C.; and Ibrahim, H. (2019). Biometric-based attendance tracking system for education sectors: A literature survey on hardware requirements. *Journal of Sensors*, Volume 2019, Article ID 74104781-25.
5. Badmus, E.O.; Odekunle, O.P.; and Oyewobi, D.O. (2021). Smart fingerprint biometric and RFID time-based attendance management system. *European Journal of Electrical Engineering and Computer Science*, 5(4), 34-39.
6. Omer, M.; and Tian, G.Y. (2018). Indoor distance estimation for passive UHF FID tag based on RSSI and RCS. *Measurement*, 127, 425-430.
7. Rennane, A.; Benmahmoud, F.; Tayeb, C.A.; Touhami, R.; and Tedjini, S. (2021). Design of autonomous multi-sensing passive UHF RFID tag for greenhouse monitoring. *Sensors and Actuators A: Physical*, 331, 112922.
8. Tu, Y.J.; Zhou, W.; and Piramuthu, S. (2018). Novel means to address RFID tag/item separation in supply chains. *Decision Support Systems*, 115, 13-23.
9. Inserra, D.; and Wen, G. (2019). Compact crossed dipole antenna with meandered series power divider for UHF RFID tag and handheld reader devices. *IEEE Transactions on Antennas and Propagation*, 67(6), 4195-4199.
10. Liu, J.; Chen, S.; Chen, M.; Xiao, Q.; and Chen, L. (2020). Pose sensing with a single RFID tag. *IEEE/ACM Transactions on Networking*, 28(5), 2023-2036.
11. Mondal, S.; Kumar, D.; and Chahal, P.A. (2020). A continuous-mode single-antenna harmonic RFID tag. *IEEE Microwave and Wireless Components Letters*, 30(4), 441-444.
12. Mondal, S.; and Chahal, P. (2019). A passive harmonic RFID tag and interrogator development. *IEEE Journal of Radio Frequency Identification*, 3(2), 98-107.
13. Islam. A.; and Karmakar, N.C. (2020). An 8×8 mm-wave LP ACMPA array for a long-range mm-wave chipless RFID tag-sensor reader. *IEEE Journal of Radio Frequency Identification*, 5(1), 53-63.
14. Kong, S.; Zhao, J.; Shi, G.; Wu, C.; Zhao, W.; and Liu, T. (2015). The design and implementation of the attendance management system based on radio frequency identification technology. *Proceedings of the International Conference on Electronic Science and Automation Control*, Zhengzhou, China, 189-192.
15. Sharma, T.; and Aarthy, S.L. (2016). An automatic attendance monitoring system using RFID and IoT using cloud. *Proceeding of the International Conference on Green Engineering and Technologies*, Coimbatore, India, 1-4.
16. Maramis, G.D.P.; and Rompas, P.T.D. (2017). Radio frequency identification (RFID) based employee attendance management system. *IOP Conference Series: Materials Science and Engineering*, 306, 012045.
17. Oo, Z.L.; Lai, T.W.; and Than, M.M. (2018). Web server base RFID attendance record system. *Proceedings of the International Conference on Recent Innovations in NanoScience and Technology*, Yangon, Myanmar, 1-21.

18. Shah, S.N.; and Abuzneid, A. (2019). IoT based smart attendance system (SAS) using RFID. *Proceedings of the International Conference on Systems, Applications and Technology*, New York, USA,1-6.
19. Khan, A.; Jhanjhi, N.Z.; and Humayun, M. (2020). Secure smart and remote multipurpose attendance monitoring system. *EAI Endorsed Transactions on Energy Web*, 7(30), 1-10.
20. Miao, Q.; Xiao, F.; Huang, H.; Sun, L.; and Wang, R. (2019). Smart attendance system based on frequency distribution algorithm with passive RFID tags. *Tsinghua Science and Technology*, 25(2), 217-226.
21. Qureshi, M.R.J. (2020). The proposed implementation of RFID based attendance system. *International Journal of Software Engineering and Applications*, 11(3), 59-69.
22. Koppikar, U.; Hiremath, S.; Shiralkar, A.; Rajoor, A.; and Baligar, V.P. (2019). IoT based smart attendance monitoring system using RFID. *Proceedings of the International Conference on Advances in Information Technology*, Chikmagalur, India, 193-197.
23. Basthomi, F.R.; Nasikhin, K.; Sa'adah, R.A.; Prasetyo, D.D.; Syai'in, M.; Rinanto, N.; Endrasmono, J.; Indarti, R.; Setiyoko, A.S.; Sukoco, D.; Herijono, B.; and Soeprijanto, A.A. (2019). Implementation of RFID attendance system with face detection using Validation Viola-Jones and local binary pattern histogram method. *Proceedings of the International Symposium on Electronics and Smart Devices*, Bandung, Indonesia, 1-6.
24. Bhagat, R. (2020). An MQTT based IoT-RFID attendance system using NodeMCU firmware: A review. *International Research Journal of Engineering and Technology*, 7(6), 1255-1259.
25. Ali, M.A.H.; and Yusoff, N.A. (2018). Development of tele-monitoring attendance system using RFID and photo-cell. *Proceedings of the International Conference on Automatic Control and Intelligent Systems*, Shah Alam, Malaysia, 83-88.
26. Magadum, A.S.; Koli, P.R; Chingale, R.N.; Zele, R.O.; Chigare. R.R.; and Rawal, C. (2022). RFID based attendance system using temperature and sanitizer. *International Journal of Research Publication and Reviews*, 3(5), 3302-3305.
27. Al-Obaidi, A.S.M.; Al-Qassar, A.; Nasser, A.R.; Alkhayyat, A.; Humaidi, A.J.; and Ibraheem, I.K. (2021). Embedded design and implementation of mobile robot for surveillance applications. *Indonesian Journal of Science and Technology*, 6(2), 427-440.