Jurnal Teknik Informatika Vol. 16 No. 1, April 2023 (1-14)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v16i1.29045.

JURNAL TEKNIK INFORMATIKA

Homepage : http://journal.uinjkt.ac.id/index.php/ti

Development of Intelligent Door Lock System for Room Management Using Multi Factor Authentication

Indra Hermawan[1], Defiana Arnaldy[2], Prihatin Oktivasari[3], Dimas Aulia Fachrudin[4], Risma Nuraini[5], Nisrina Tsany Sulthanah[6]

[1,2,3,4,5,6] Informatics and Computer Engineering Department, Jakarta State Polytechnic
[1.2,3,4,5,6] Jl. Prof. DR. G.A. Siwabessy Kampus, Kukusan, Kecamatan Beji, Kota Depok, Jawa Barat 16425, Indonesia

ABSTRACT

Keys play an important role in a security system. Most room door security systems still use conventional door locks, especially in room management applications. Room management with conventional keys has several problems, such as long key searches for each loan, and easily lost and duplicated. Therefore, the authors propose a door lock system that is capable of automating room door control and supervision via the internet network. Research that has been done previously uses single factor authentication (SFA). This technology is less secure for the authentication process for parties/users who want to access the room. This research proposes that a system with multi factor authentication (MFA) aims to protect the system from unauthorized users. The MFA concept will be implemented by validating the physical token on the RFID card and the security code or PIN entered by the user. MFA also increases the level of security and allows the implementation of identification, verification and authentication to ensure user authority. The results of this study are system prototypes that can work well, the system can be realized using backend services using expressjs and data stored in cloud services that can be accessed anytime and from anywhere by the client. The system can also be realized using a combination of nodejs, expressjs, PostgreSQL and esp32 client technologies. The validation process at check-in can occur within 207ms. The server is reliable enough to execute 500 requests simultaneously. Overall, the system has worked well for room management.

Keywords: *smart door lock, nodejs, expressjs, esp32, multi factor authentication*

Article:
Accepted: April 21, 2023
Revised: March 14, 2023
Issued: April 30, 2023

***Correspondence Address:**
indra.hermawan@tik.pnj.ac.id

1. INTRODUCTION

Room management encompasses various practices to ensure room security and restrict access to authorized individuals [1]. Its primary objective is to safeguard valuable assets and deter unauthorized entry into the room. Managing a large number of rooms can present significant challenges for room management. As the number of rooms being managed increases, the complexity of tasks such as limiting who can access the room also increases. Room management with conventional locks has limitations such as each room has one key, and the key cannot be used in other rooms. The number of keys to access each room will increase the risk of lost keys so that the room cannot be accessed [2]. Another problem arises when finding the right key for a room, the key search can take a long time. Conventional room management also requires someone to keep watch and provide the keys needed, humans have limits to stay awake and active, this will impact the limited time that can be used to use the room. Research [3] highlights the inadequacy of conventional door locks in effectively identifying unauthorized access. Furthermore, it reveals that such lock systems are prone to reliability issues, which can compromise the overall security of the managed room. In addition, research [1][4] further emphasizes a significant weakness of conventional door systems: the vulnerability of keys to unauthorized duplication by unscrupulous individuals. This susceptibility raises concerns about the reliability and integrity of the room management system, potentially compromising the security measures in place. Problems with conventional doors can be overcome with an automated smart door lock system. Smart door lock systems have higher security, are adaptive, flexible, and reliable in various conditions of use (can be used in homes, offices, campuses, health facilities, etc.) [5]. Several studies have been carried out to make this smart door lock system.

Table 1. Previous research

Journal	Devices	Description
[4]	PIR motion sensor, MG995 Metal Gear Servo, ESP32 Camera, NodeMCU ESP8266, Blynk application	Doors can be opened using virtual keys in the Blynk app. If the motion sensor detects movement, the video streaming process begins.
[6]	Raspberry Pi, Webcam, Solenoid Lock, 5volt buzzer, Relay Module, power supply 5 V dan 12 V, Open CV, Python	The camera takes a face image, the Raspberry Pi will validate and then take action, namely opening or closing the door
[7]	RFID (RDM6300), Arduino, Ethernet Shield,	Arduino reads RFID data, sends data via ethernet cable, and then stores the card ID in the database.
[8]	Raspberry Pi, RFID reader, keypad, LCD, electromagnet, solenoid EM, power supply 12v, relay, DC power jack	RFID-based door lock system with OTP-based technology to provide high-security solutions for households. In this device, OTP is generated for door access, and this OTP will expire after the given expiration time.

Table 1 shows the results of previous studies. Research [4], using single factor authentication (SFA) by only using a virtual button on the Blynk application, this technology is deemed incapable of performing user management functionality. Virtual buttons cannot authenticate parties/users with access permissions to the room or users who do not have access permissions. Research [6] using facial recognition technology will face serious problems if the number of users continues to grow. In this study the testing process was carried out on 14 samples of user faces, of course this cannot represent the number of real users in the field which can reach thousands of users. Problems that occur can be in the form of lengthy authentication processes or authentication process errors. An accuracy rate of 94% can also be a threat if the user gives access to the wrong person.

Research [7] [8] uses RFID technology to authenticate. The card validation process works relatively quickly and precisely compared to the face recognition process. Research [8] added the OTP code as an option to access the room. Research [7] has not implemented a room management process, the system only keeps logs of card use without special access settings into the room. Based on the things that have been stated above, the formulation of the problem that is used as the focus of the irrigation system being built:

a) How to come up with a smart door lock system that can do room management
b) How to present a smart door lock system that can work quickly.

2. METHODS

2.1. Research Phases

To realize a smart door lock system that can solve conventional lock problems, a smart door system must have characteristics such as reliability, capable to verify and limit visitor access, and have interactive user interface and easy to operate.

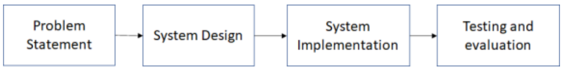

Figure 1. Research Phase

To achieve these objectives, this research will be carried out in several stages as shown in the Figure 1.

2.2. Problem Statement

An in-depth literature study showed that conventional door lock systems suffer from significant limitations, including compromised security and the inability to authenticate users effectively. Thus, this research endeavors to address these shortcomings by prioritizing the development of smart door systems capable of proficient room management. The primary objectives encompass restricting access to room visitors and ensuring the systems operate swiftly and efficiently. By doing so, the research aims to provide enhanced security and seamless functionality for optimal room management.

2.3. System Design

The smart door lock system will have the ability to perform room management. Room management will be carried out based on access cards. The access card will be read using the Radio Frequency Identifier (RFID) module. ID card data will be stored in the database. Administrators can grant access to cards to access certain rooms. *Figure 2* shows the architecture of the smart door lock system.

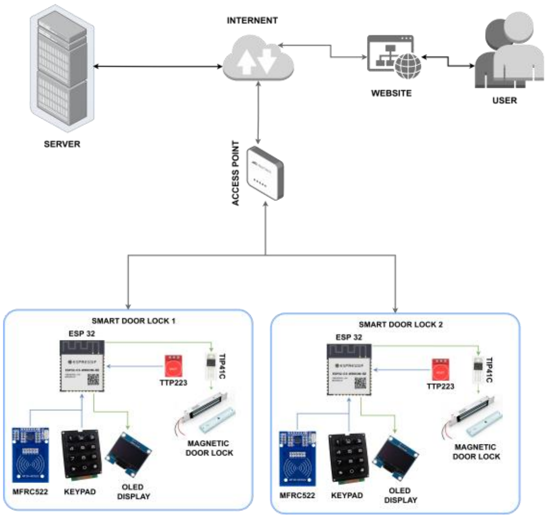

Figure 2. System architecture

The smart door lock system consists of 3 main modules. The first module is a sensor module. This module has the ability to read RFID cards and receive input from the keypad, besides that, this module provides information about the validation status through the display. The second module is the power supply module, this module functions to lower the voltage and regulate the current flowing to the magnetic door lock, this module has a touch button, which functions to send a signal to the microcontroller to cut off the power to the magnetic door lock. The last module is the actuator module which contains a magnetic door lock, which holds the door so that it cannot be opened. The list of all components is shown inTable *2*. The smart door lock system has two modes. The first mode is the check-in mode, and the second mode is the registration mode. Check-in mode is the default mode when a new device is turned on. This mode serves to authenticate users who want to enter the room. While the registration mode is a mode for registering a user card into the system. To change mode, the administrator needs to change mode and enter the device pin. When the pin entered is correct, the device will change mode.

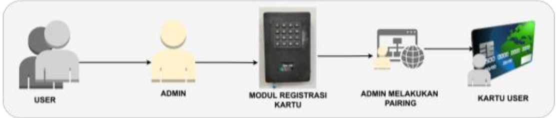

Figure 3. Card registration process

Figure 3 shows the card registration process. Cards that can be registered are cards with a radio chip at a frequency of 13.56Mhz, the user needs to see the administrator to register the card. The administrator must turn the device into registration mode by entering the pin. After the device mode changes, the user is asked to enter six pins, and then the administrator attaches the user's card to the device. The device will provide information on whether the card was registered successfully or not. The card failed to be registered if the card has been successfully registered before.

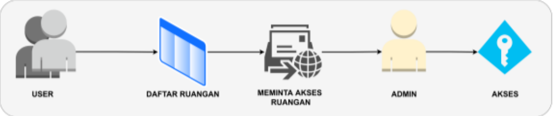

Figure 4. The process of requesting room access

Figure 4 shows the process of setting access to the room. Users can see a list of available rooms and then send a request to access the room. Admin can see a list of users who request access to the room. Admin can reject or approve user requests. If the admin approves the user's request, the user can access the previously requested room.

The user's card has a few security features. This feature is available to ensure the security of the user's card. This feature is multi-factor authentication (MFA). MFA is a system that uses two or more different factors to authenticate a user, MFA provides two ways of verification, one of which uses a physical token, such as a card, and the other is usually memorized, such as a security code [9]. MFA aims to protect the system from unauthorized users, MFA also increases the level of security and allows the implementation of identification, verification, and authentication to ensure user authority, indirectly reducing the risk of malicious attacks [10]. In this research, MFA is a feature that makes access to a room require a pin and an access card. *Figure 5* shows the process of entering into a room. If the user activates MFA, the user must enter the pin and tap the access card. If the user doesn't activate MFA, they only need to tap the access card.

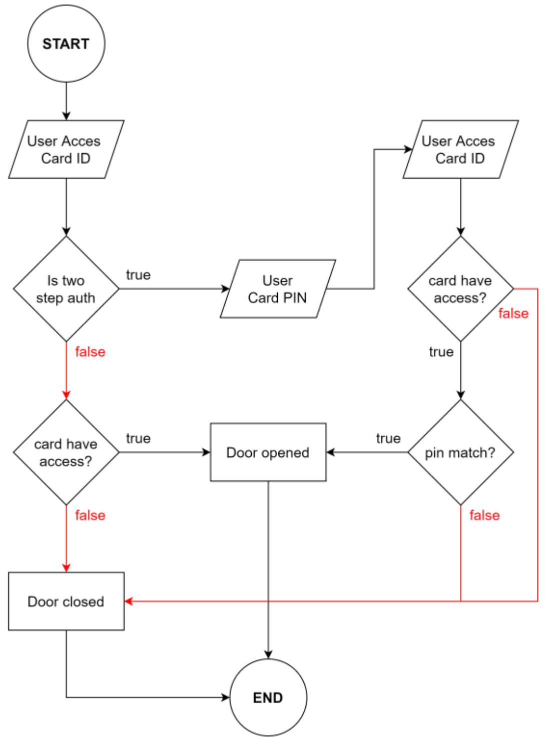

Figure 5. Flowchart during the check-in process

2.4. Hardware Design

The smart door lock system being developed will use the IoT paradigm. IoT is a new technology development that allows a device to be connected to a global network and devices that are able to interact with each other [11]. This IoT technology has been used in various applications such as smart homes[12], smart farming [13], smart irrigation [14], smart trash bin [15], surveillance system [16] and other. This study will use a device-to-cloud (D2C) communication strategy. D2C is an architecture that provides a two-way connection between IoT devices and cloud services to send information or send instructions [17]. The smart lock system will send data using HTTPS technology

Table 2. List of hardware used

Module	Device	Function	Specificaton
Sensor	ESP 32	The main microcontroller is to process information and communicate to the server. Set the data trigger that serves to drain or break the current to the actuator.	Equipped with WIFi 802.11, Bluetooth version 4.2, and various peripherals. This board has two versions namely 30 GPIO and 36 GPIO. The ESP 32 board has a USB to UART interface which is easy to program with programs such as the Arduino IDE. Board resources can be provided via the microUSB connector [18].
	MFRC522	Module for reading access cards.	This RFID Reader has an operating voltage of 3-26mA/DC 3.3V with a frequency of 13.56MHz. MFRC522 has a maximum data transfer rate of 10 Mbit/s with the SPI protocol. This reader can read or accept the following cards: mifare1 S50, MIFARE DESFire, mifare Pro, and mifare1 S70 MIFARE Ultralight [19].
	KEYPAD	Input board to enter user pin.	Keypad 4x4 has a maximum rating: 24 Vdc, 30 mA. The interface is 8 pin to access the 4x4 matrix keypad. Operating temperatures range from 32 to 122 degrees F (0 to 50 degrees Celsius). This keypad measures: 6.9 x 7.6 cm
	PCF8574	I2C module for the keypad so as not to use too many pins on the microcontroller.	8 bit I/O expander for I2C bus and is a shift register. PCF8574 is made of silicon CMOS circuit consisting of an operating voltage of 5V dc, 2 control channels, pins NO, NC, VCC, and GND [20].
	OLED DISPLAY SSD1306	Displays information such as device mode, check-in information, wifi connection	This display has a 128x64 dot matrix panel resolution, on-chip for COG and COF, row remapping and column remapping. This OLED display also has an operating temperature range of -40°C to 85°C, programmable frame rates and multiplexing ratios. The display matrix has a maximum OLED drive voltage of 15, a maximum segment source current of 100µA, a maximum sink current of 15mA, and 256 steps of current control brightness contrast [21].
Power Supply	TTP223	Touch sensor that will send information to the microcontroller to cut off the current to the actuator.	This touch sensor has an operating voltage of 2.0V-5.5V, provides low power mode, direct mode switch mode with pad option (TOG pin), and lifetime auto-calibration. In low power mode, the recalibration period is normally about 4s and about 16s from key release. The TTP223 has a maximum response time of 220ms in low power mode and its sensitivity can be adjusted according to the capacitance (0-50pF) outside [22].
	TIP41C	Transistor that functions as a relay to disconnect or flow current to the actuator.	The specifications of the TIP41C transistor include the TO-220 package type with the NPN transistor type which has a maximum collector current of 6A or 6000mA. This transistor has a maximum collector-emitter voltage of 100V, a maximum collector-base voltage of 100V, and a maximum transmitter base voltage of 5V [23].

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v16i1.29045.

Table 2 continued…

Module	Device	Function	Specificaton
	MP2307	Module to reduce the power of the electric current.	The MP2307 device integrates a 100mΩ MOSFET that provides 3A of continuous load current at a wide operating input voltage of 4.75V to 23V. Current mode control provides fast transient response and cycle-by-cycle current limit. Adjustable soft-start to prevent inrush current when power on and in off mode, supply current drops below 1µA. This device, available in an 8-pin SOIC package, provides a very compact system solution with minimal dependence on external components [24].
Actuator	MAGNET IC DOOR LOCK	Holding the door from opening	The 600 Lbs magnetic door lock has specifications: the supply voltage is 12Vdc, 24Vdc and has a current draw of 500mA at 12Vdc and 250mA at 24Vdc at 20°C. The depth of power of this magnetic door lock is 600 lbs or 280 kg with a size of 250 x 42 mm with a weight of 2kg and humidity 0 to 95% without condensation. The ambient working temperature is 10 to 55°C and the lock surface temperature is 20% when the power is on [25].

Figure 6 shows a schematic circuit of the electronic devices in the smart door lock system. The module works using an AC to DC 12v 3A adapter. Great power is required to activate the magnetic door lock. ESP 32 requires 5v of power to operate. The MP2307 dc to dc step-down module is used to reduce the 12v voltage to 5v. Other devices use the 3.3v pin on the ESP 32 to work. The TTP 223 touch sensor module gets power directly from the MP2307, the incoming power is regulated again by the AMS1117 3.3v module, which will reduce the 5v power provided by the MP2307.

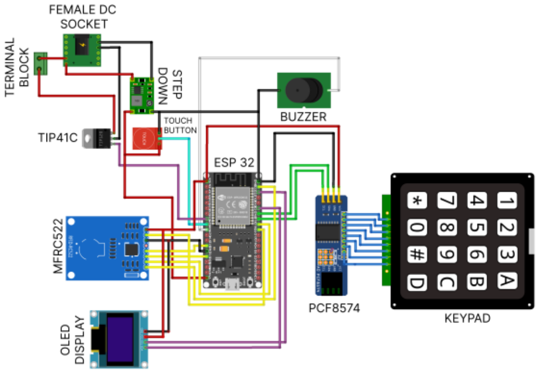

Figure 6. Diagram of smart door lock system

To protect electronic circuits from dust, water splashes or other debris, the smart door lock system has a hard case. The hard case is designed using fusion360 and adjusts the size of the electronic module used. All hard cases are printed with PLA plastic using a 3D printer.

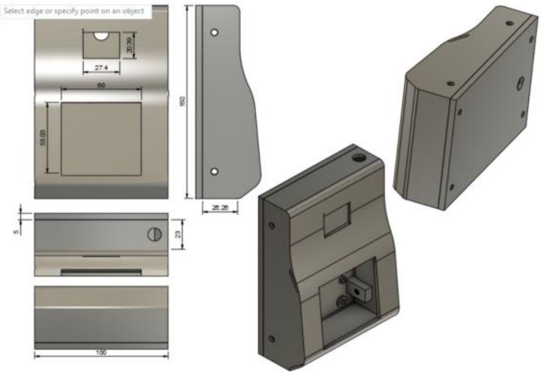

Figure 7. Card reader case design

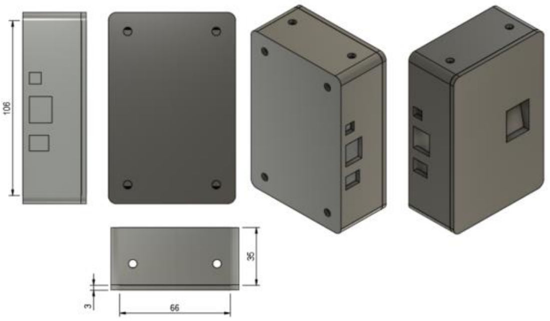

Figure 8. Design of PSU and touch sensor

There are two hard cases built for the sensor module and power supply. *Figure 7* shows the sensor module with a hardcase size of 100mm × 160mm with a thickness of 26mm to 40mm. Whereas *Figure 8* shows the hard case for the power supply having a size of 106mm × 55mm with a thickness of 38mm.

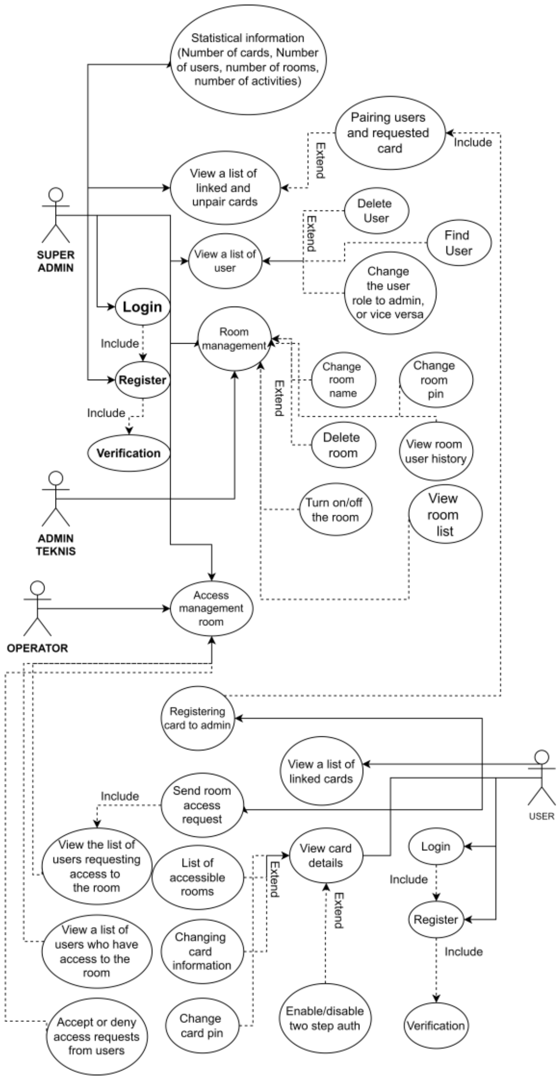
Statistical information (Number of cards, Number of users, number of rooms, number of activities)
Pairing users and requested card
Include
Extend
View a list of linked and unpair cards
Delete User
Find User
SUPER ADMIN
View a list of user
Extend
Change the user role to admin, or vice versa
Login
Room management
Include
Register
Change room name
Change room pin
Extend
Include
Delete room
View room user history
Verification
Turn on/off the room
View room list
ADMIN TEKNIS
Access management room
OPERATOR
Registering card to admin
View a list of linked cards
USER
Include
Send room access request
View the list of users requesting access to the room
List of accessible rooms
View card details
Extend
Login
Include
Extend
Register
View a list of users who have access to the room
Changing card information
Include
Accept or deny access requests from users
Change card pin
Enable/disable two step auth
Verification

Figure 9. Activity diagram

2.5. Software Design

a. Backend service design

The backend service is made with JavaScript and the help of the Expressjs web library. Several studies have proven that JavaScript and Express are fast services and can run with limited resources. Research [26], shows JavaScript to be the programming language of choice for small-scale projects to master Java and Python. JavaScript can also execute tasks faster than these two languages, even though it can execute tasks quickly, the Java programming language is more stable to use in various conditions. Research [27] shows that Expressjs can execute tasks and provide a better response time compared to ASP.NET, JavaScript is also far more optimal in utilizing available hardware. In the same study, Expressjs uses 3 times less memory and uses two times more CPU low.

Figure 9 shows the activity diagram of the backend service. Users will be divided into four roles with different tasks. Role Users can only perform basic functions such as requesting room access and viewing a list of rooms that can be accessed. The Operator role is responsible for performing basic administration tasks, such as linking users and cards and granting or removing user access to rooms. The Technical Admin role ensures that the hardware is connected perfectly to the cloud service. This role creates room data and links it with available hardware data. The Super Admin role can perform tasks performed by operators and technical admins; the super admin can change each user’s role.

b. Firmware Design

The firmware is written in C language. The C programming language was chosen because of its efficiency and speed making it suitable for use in embedded devices [28]. ESP 32 allows firmware updates wirelessly, this allows the device to be implemented in a native environment, when there are errors or bugs, updates or patches can be done easily using wireless technology. Firmware is designed to read RFID input along with input from keypads or touch sensors. The firmware will also control the OLED display and the relay, which is responsible for flowing or muting the electric current to the actuator. When the device is turned on, the microcontroller will check the EEPROM, whether the device already has an ID. If not, the device will make a request to the server to request a device ID. If the device already has an ID, the device will make a request to the server to ensure the device ID is still active, if the device ID is no longer active, the device will again request a new ID. The device ID will later be used to link hardware with room data. Each room has only one device.

After initializing the smart door lock system, it will display a welcome screen. When the device is idle, the user can enter a pin or tap an access card. The device will send a request containing the room ID, card ID, along with the pin. If the user activates two step authentications, the device will send the data to the cloud service, if the response given is code

200, then the user can enter the room and the display gives information that the room opened successfully. Code 500 is given if the user does not have access to the room, the display will provide information that the room has failed to open, while code 401 indicates the user's pin is incorrect, the display will ask the user to enter the correct pin. When the back end service give code 200, ESP32 will send a trigger signal to TIP41C to cut off current to the actuator.

c. Database Design

The database used is PostgreSQL. PostgreSQL is the best choice for small to medium sized projects. Tests conducted [29] showed that PostgreSQL had the best performance for processes involving 100-500 insert queries, requiring only an average of 383.22ms, outperforming MySQL (470.25ms) and MongoDB (612.56ms) [29].

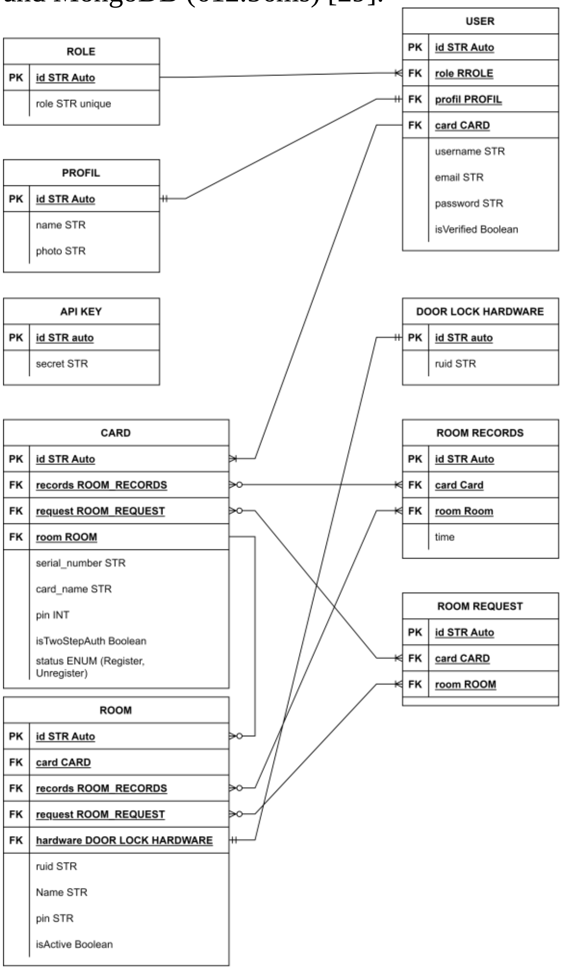

Figure 10. Database design

Figure 10 shows the design of the database structure, the databases will be related to one another. The API KEY table is responsible for storing API data, this data will be used to validate connections from the hardware, if the API ID and API KEY sent are not recorded, requests made by the device will not be responded to.

The user table has bindings to other tables such as roles and profiles. The user table is connected one to many with the card table, each user can have several cards. These cards will have a relationship to the rooms that can be accessed, this structure is represented by a many to many relationships, which allows one card to be used to access several rooms at once. Before the user card can access the room, the user must send a request to the administrator and the data will be temporarily stored in the REQUEST ROOM, after the administrator takes action the request data will be deleted from the table. The ROOM (room) table will have a one-to-one relationship with DOOR LOCK HARDWARE (hardware). This is done for the purpose, if the hardware is damaged then the room can be linked with replacement hardware, so that card data that has access to the room does not need to be re-registered. When the user taps the access card, the process will be recorded and entered into the ROOM RECORDS table.

2.6. Testing and Evaluation

a. Testing

The testing phase is crucial to ensure the system functions optimally and achieves its intended goals. Multiple tests are conducted to evaluate different aspects of the system's performance. The initial test focuses on assessing the hardware capabilities, while the subsequent test aims to evaluate the software's functionality.

Hardware testing comprises two sections. Firstly, the durability of the hardware is examined in its intended environment to verify its reliability. Secondly, a stress test is conducted to evaluate the long-term performance of the microcontroller, particularly its ability to handle repetitive tasks. These tests collectively ascertain that the system can effectively execute room management operations.

Software testing is conducted through four distinct schemes. These schemes encompass testing user registration, card registration, room access requests, and the check-in process. The objective is to evaluate the system's speed and responsiveness in performing these essential functions. The system's efficiency and responsiveness can be validated by conducting comprehensive software testing, ensuring seamless operation throughout the room management process.

b. Evaluation

During the hardware testing process in its original environment, one of the test parameters collected is the device's operational duration without encountering any issues. This parameter provides valuable insights into the hardware's reliability and stability over an extended period of usage.

On the other hand, the stress test performed on the hardware focuses on evaluating its performance in terms of the time required to complete a specific process. In this case, the process under scrutiny is the request process sent to the server. The hardware's efficiency and responsiveness can be determined by assessing the time taken to execute this process under stress conditions, providing crucial information for optimizing its performance.

All tests conducted on the cloud service software aim to evaluate its performance, specifically determining the system's responsiveness. The primary data collected during these tests is the response time, which provides valuable insights into the system's speed. By analyzing the response time data, it becomes possible to assess whether the system functions quickly and responsively. A lower response time indicates efficient performance, demonstrating that the system can promptly handle user requests and deliver timely responses. This data plays a crucial role in gauging the effectiveness of the cloud service software and ensuring optimal user experience in terms of speed and responsiveness.

2.7. Testing Environment

The hardware testing was conducted in the real-world environment of the embedded lab room at the Jakarta State Polytechnic. This setting allowed for a comprehensive evaluation of the hardware's performance and durability under conditions similar to its intended use.

In contrast, the cloud service was tested by simulating high user traffic using Python scripts. This approach enabled the assessment of the cloud service's performance and scalability when subjected to a heavy workload. By simulating high user traffic, the system's response time, stability, and ability to handle concurrent requests can be effectively measured and evaluated.

The server that will be used in the test is a computer with an operating system Windows 10 platform build 19044, with CPU i5-8265U CPU @ 3.80GHz (4 cores, 8 threads) with 4GB RAM.

3. RESULTS AND DISCUSSION

There are several test schemes performed on hardware. The first scheme is carried out to test the capability and reliability of the hardware. This test is carried out by installing a prototype for one month in a real environment. The second scheme is to test performance and reliability, testing is done by conducting a stress test by sending a number of data in several batches.

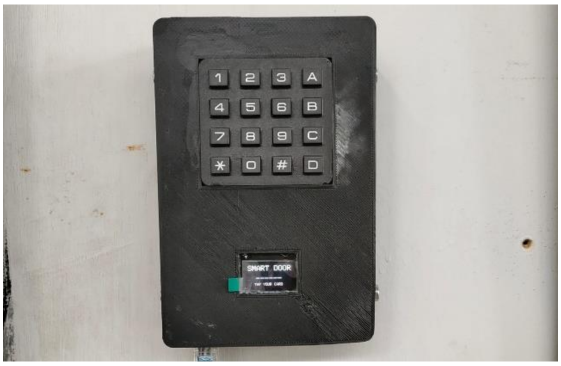

Figure 11. Card reader module

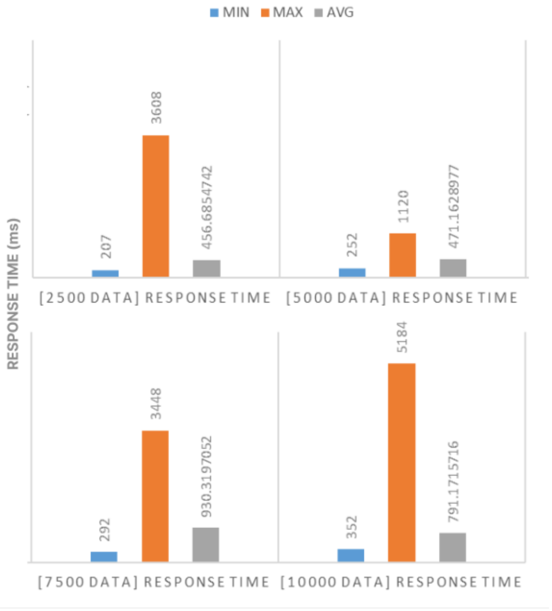

Figure 12. ESP32 stress test results

3.1. Hardware Testing

The first test scheme takes place from October 1, 2022 to November 10, 2022. *Figure 11* is an implementation of the device in a real environment. During the testing period the device could work well. All functionality works well without any problems. A 12v 3a power supply powers the device. The power is divided into the magnetic door lock, and some of the power is reduced to 7v and is channeled to the ESP 32 using a UTP cable. With 7v power, the ESP 32 can still work normally without experiencing a drastic increase in temperature.

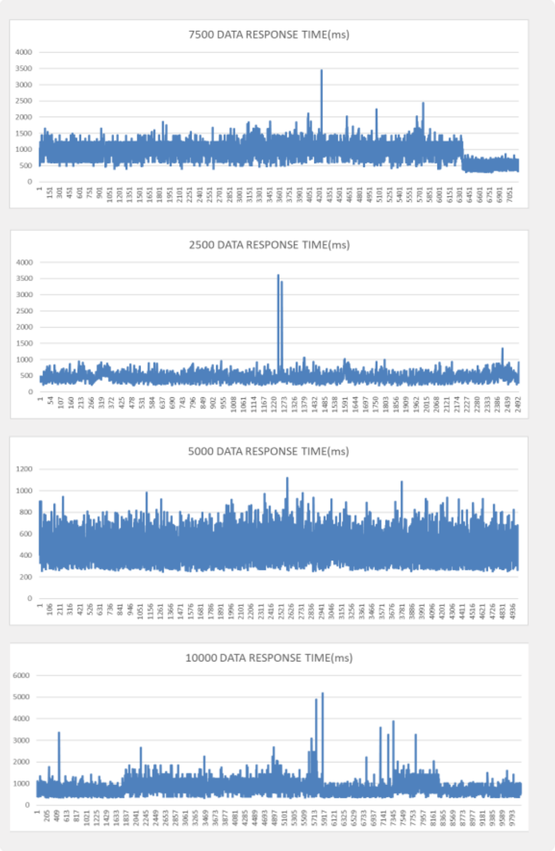

Figure 13. Comparison of response times

The second testing scheme is to do a stress test. Each test batch contains 2500, 5000, 7500, and 10000 data stored in json format. The data is stored on the computer and sent via a serial cable to the ESP 32. The ESP 32 will make a request to the server. The request is sent in the form of a validation request whether the card has access to the room, the response time for program execution will be sent back to the computer and recorded in a *.xlsx format file. *Figure 12* shows that the minimum response time continues to increase with the amount of data sent. Meanwhile, the maximum response time has increased from 5000 to 10000 data. The average response time also increases with the amount of data that must be sent.

Figure 13 shows that the response time for each batch is quite stable. In a batch of 2500 data, there are several requests that have response times above 3000ms. Batch 10000 Data is batch with the most data fluctuations. Overall, ESP 32 has stable performance. Although sometimes there are fluctuations, ESP 32 can return to stability on subsequent requests.

3.2. Backend Service Testing

The backend service in the intelligent door lock system is one of the vital systems. The backend service will save the card data and validate if the card can open the room. Therefore, backend service testing is needed to determine performance, capability, and reliability. Testing is done by sending many requests to the server. Requests will be sent via scripts written in python and utilizing threading technology. The more threads that are used will simulate the number of users who send requests to the server. The testing process will be carried out with 2500, 5000, and 7500 data and using a schema of 50, 250, 500 threads. The testing process will be carried out on the user registration process, card registration, request for access to the room, as well as the check-in process into the room.

1. User registration process

The user registration process is the earliest process in using the system. This process requires username, email and password data. Data is generated randomly. Data with the same email or username, the registration process will fail. The test results can be seen in the Table 3.

Table 3. Test results of the registration process

Data Size	Thread Size	Total Success	Min	Max	Avg
2500	50	2499	2497	6010	3800
	250	2499	401	29512	21292
	500	683	383	34517	21809
5000	50	5000	801	6154	4520
	250	5000	1080	30011	21021
	500	451	441	43361	26069
7500	50	7500	1374	5866	4331
	250	7496	9310	45024	22732
	500	336	553	27809	15104

Table 3 is the result of testing the registration process. The average response time results, on each of the same thread with different amounts of data, are relatively the same and have not experienced a significant increase. A significant increase occurs when the number of threads is increased. At the same amount of data, the increase in average time experienced a very high spike, in batch 2500 data, the response time of 250 threads increased 6 times compared to 50 threads.

2. Card registration process

The card registration process is carried out via an ESP 32 device. This test will simulate the registration process will be simulated using a python script. The number of registered cards depends on the number of users that can be registered.

Table 4. Test results for the card registration process

Data size	Thread size	Total success	Min	Max	Avg
2500	50	2499	654	9619	6378
	250	2499	940	24892	19580
	500	430	928	32674	20775
5000	50	5000	685	6622	4523
	250	5000	6723	21871	17100
	500	368	12145	31502	19317
7500	50	7496	166	5344	4056
	250	7496	1320	20605	17460
	500	416	11802	30190	18544

Table 4 test results show that the test results are more or less the same as the user registration test. Adding the amount of data tested does not really affect the minimum response time. Meanwhile, the more threads used, the higher the minimum response time. In a batch of 5000 data, the minimum response time on 50 threads is 685ms increasing to 6723ms on 250 threads and increasing 2 times again to 12145ms on 500 threads.

3. The process of obtaining access to the room

Users can request access to certain rooms. Users can perform these actions through the website. The simulation that will be carried out will represent the number of users requesting a room at one time.

Table 5. The results of testing the access request process

Data Size	Thread Size	Total Success	Min	Max	Avg
2500	50	2499	72	2360	534
	250	2499	223	3118	2172
	500	683	1394	3891	2873
5000	50	5000	53	2122	593
	250	5000	151	4863	2601
	500	1055	1782	4717	2963
7500	50	7496	50	1776	396
	250	7496	394	5487	2454
	500	1406	754	3891	2387

Testing the room access request process is shown in Table 5. The results obtained are almost similar to the tests carried out before. The maximum response time does not increase dramatically when the amount of data increases but increases dramatically when the number of threads increases. This can be seen in the 7500 data batch, the maximum time continues to increase from 1776 on 50 threads to 5487ms when 250 threads. The maximum time then decreases at 500 threads to 3891ms. This decrease occurred because the server could not respond to all incoming requests, only a few requests were made by the server.

4. Check-in process

The check-in process is a process when the user wants to enter the room. The device will read the RFID card and send the data via wifi to the server. The server will respond whether the user is allowed to enter the room or not. The testing process will be simulated using a Python script.

Table 6. The results of the check-in process

Data Size	Thread Size	Total success	Min	Max	Avg
2500	50	2499	798	8369	5780
	250	2499	600	28966	25687
	500	683	274	45538	29081
5000	50	5000	2685	8592	6204
	250	5000	2497	57146	37460
	500	502	286	34810	23085
7500	50	7496	3751	14438	7816
	250	7496	8444	47339	38787
	500	763	314	36001	22951

Table 6 is the result of the check-in process test. The test results still show similarities with the previous test. The number of threads will increase the response time. In addition, the large number of threads will reduce the success of the request process. The decrease in request success is caused by the inability of the server to respond to every request sent. The server is unable to respond to all requests because the server is running on one script on the computer. The load is not distributed to other servers. To overcome this the load of requests received by the server must be distributed. This technology is known as load balancing.

CONCLUSION

Based on the results of the tests, it can be concluded that the smart door lock system was successfully built and tested. The findings demonstrate that:

1. A room management system with MFA can be implemented using a backend service built with expressjs and an ESP32 microcontroller. Data stored in the cloud allows multiple clients to access the data from anywhere.
2. A fast smart door lock system can be created using a combination of nodejs, expressjs, PostgreSQL, and ESP32 client technologies. The validation process at check-in can be completed within 207ms, and the server can handle up to 500 requests simultaneously.

Overall, the study shows that the system can effectively manage rooms and provide reliable security for users. Further research can focus on improving the system's scalability and expanding its features for broader applications.

ACKNOWLEDGEMENT

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Politeknik Negeri Jakarta grant numbers B.531/PL3.18/PT.00.06/2022.

REFERENCE

[1] IEEE Control Systems Society. Chapter Malaysia and Institute of Electrical and Electronics Engineers, "Development of Web-Based Smart Security Door Using QR Code System," in *2020 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS 2020)*, 2020, pp. 13–17.

[2] F. As *et al.*, "DESIGN AND CONSTRUCTION OF A SMART DOOR LOCK WITH AN EMBEDDED SPY-CAMERA," 2021. [Online]. Available: https://www.researchgate.net/publication/354872757

[3] Y. Hasan, Abdurrahman, Y. Wijanarko, S. Muslimin, and R. Maulidda, "The Automatic Door Lock to Enhance Security in RFID System," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, May 2020. doi: 10.1088/1742-6596/1500/1/012132.

[4] U. Annisa Binti Norarzemi *et al.*, "Development of Prototype Smart Door System With IoT Application," *Progress in Engineering Application and Technology*, vol. 1, no. 1, pp. 245–256, 2020, doi: 10.30880/peat.2020.01.01.027.

[5] H. R. Abdulshaheed, H. H. Abbas, I. Al Barazanchi, and W. Hashim, "CONTROL AND ALERT MECHANISM OF RFID DOOR ACCESS CONTROL SYSTEM USING IOT," *Glosas de Innovacion aplicadas a la pyme*, pp. 269–285, 2022, doi: 10.17993/3ctecno.2022.

[6] F. Azmi, I. Fawwaz, and R. Anugrahwaty, "Smart Door System using Face Recognition Based on Raspberry Pi," *JURNAL INFOKUM*, vol. 10, no. 1, 2021, [Online]. Available: http://infor.seaninstitute.org/index.php/infokum/index

[7] D. Yulianto Wijaya and A. Yulianto, "Prototype of Smart Door Using RFID Technology with Internet of Things (IoT)," *Conference on Management, Business, Innovation, Education and Social Science*, vol. 1, no. 1, 2021, [Online]. Available: www.arduino.cc

[8] D. Aswini, R. Rohindh, K. S. Manoj Ragavendhara, and C. S. Mridula, "Smart Door Locking System," in *2021 International Conference on Advancements in Electrical,*

Electronics, Communication, Computing and Automation, ICAECA 2021, Institute of Electrical and Electronics Engineers Inc., 2021. doi: 10.1109/ICAECA52838.2021.9675590.

[9] M. Rusdan and D. T. H. Manurung, “Designing of user authentication based on multi-factor authentication on wireless networks,” *Journal of Advanced Research in Dynamical and Control Systems*, vol. 12, no. 1, pp. 201–209, 2020, doi: 10.5373/JARDCS/V12I1/20201030.

[10] A. J. Mohammed and A. A. Yassin, “Efficient and flexible multi-factor authentication protocol based on fuzzy extractor of administrator’s fingerprint and smart mobile device,” *Cryptography*, vol. 3, no. 3, pp. 1–222, Sep. 2019, doi: 10.3390/cryptography3030024.

[11] I. Hermawan, D. Arnaldy, M. Agustin, M. F. Widyono, D. Nathanael, and M. T. Mulyani, “SISTEM PENGENALAN BENIH PADI MENGGUNAKAN METODELIGHT CONVOLUTIONAL NEURAL NETWORKPADA RASPBERRY PI 4 B,” *Jurnal Teknologi Terpadu*, vol. 7, no. 2, pp. 120–126, 2021.

[12] Purnawan Peby W. dan Rosita Yuni, “Engineering of Smart Home System Using NodeMCU Esp8266 Based on Telegram Messenger Communication,” *Techno.COM*, vol. 18, no. 4, pp. 348–360, 2019, [Online]. Available: http://publikasi.dinus.ac.id/index.php/technoc/article/view/2862

[13] P. A. Rezeki, F. Dewanta, and S. Astuti, “Rancang Bangun Smart Farming untuk Observasi Pertumbuhan Tanaman Kangkung dengan Dukungan Teknologi Sonic Bloom,” vol. 8, no. 1, pp. 50–59, 2022.

[14] I. Hermawan, D. Aulia Fachrudin, A. Setiawan, and N. Tsany Sulthanah, “Rancang Bangun Sistem Irigasi Cerdas Menggunakan Metode Fuzzy Rule-Based untuk Otomatisasi Pintu Air dan Pendeteksian Endapan,” *Jurnal Komputer Terapan*, vol. 8, no. Vol. 8 No. 1 (2022), pp. 1–11, 2022, doi: 10.35143/jkt.v8i1.5253.

[15] R. Hanafi, “RANCANG BANGUN TEMPAT SAMPAH PINTAR DAN MOBILE MENGGUNAKAN SISTEM INFORMASI BERBASIS TEKNOLOGI INTERNET OF THINGS,” 2021.

[16] I. Hermawan *et al.*, “Low-cost Surveillance System using Smartphone and Raspberry Pi4 Based on Real Time Streaming Protocol,” *2022 5th International Conference on Computer and Informatics Engineering, IC2IE 2022*, pp. 106–110, 2022, doi: 10.1109/IC2IE56416.2022.9970093.

[17] A. Souri, A. Hussien, M. Hoseyninezhad, and M. Norouzi, “A systematic review of IoT communication strategies for an efficient smart environment,” *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 3, Mar. 2022, doi: 10.1002/ett.3736.

[18] I. Hermawan, A. Kurniawan, M. Agustin, D. A. Fachrudin, N. Tsany, and A. Setiawan, “Pengendalian Sistem Irigasi Berbasis Komunikasi Radio Full Duplex dengan Algoritma Decision Tree,” vol. 9, no. 1, pp. 13–26, 2023.

[19] R. Laksmana Singgeta, P. D. Manembu, and R. G. Sangkay, “IMPLEMENTASI TEKNOLOGI RFID PADA DISPENSER AIR MINUM,” *Jurnal Elektro*, vol. 12, no. 1, pp. 23–32, 2019.

[20] A. Dey, S. Mukherjee, and K. Palit, “Pump Controlling Solution using Smart Technology with Audio Notification,” in *Proceedings of the Fifth International Conference on Communication and Electronics Systems*, 2020, pp. 233–237.

[21] A. Christopher and Y. M. Dinata, “Rancang Bangun Sistem Pemantauan Jarak Jauh Denyut Nadi, Saturasi Oksigen, dan Suhu Tubuh pada Orang Sakit di Rumah,” *JUISI*, vol. 08, no. 01, pp. 1–11, 2022.

[22] Tontek, “TTP223-BA6 TTP223N-BA6.”

[23] FAIRCHILD SEMICONDUCTOR, “TIP41C Series,” 2000.

[24] Monolithic Power Systems, “MP2307 Synchronous Rectified Step-Down

Converter,” 2006. [Online]. Available: www.MonolithicPower.com

[25] RS PRO, “Magnetic Door Locks.” [Online]. Available: https://uk.rs-online.com/

[26] L. R. Abbade, M. A. A. da Cruz, J. J. P. C. Rodrigues, P. Lorenz, R. A. L. Rabelo, and J. Al-Muhtadi, “Performance comparison of programming languages for Internet of Things middleware,” *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 12, Dec. 2020, doi: 10.1002/ett.3891.

[27] Oliver Karlsson, “A Performance comparisonBetween ASP.NET Core andExpress.js for creating Web APIsMAIN FIELD:Computer ScienceAUTHOR:Oliver KarlssonSUPERVISOR:Peter Larsson-GreenJ˙ONK˙OPING:July 2021,” JONKOPING UNIVERSITY, JONKOPING, 2021.

[28] I. Hermawan, D. A. Fachrudin, A. Setiawan, and N. T. S. Sulthanah, “Rancang Bangun Sistem Irigasi Cerdas Menggunakan Metode Fuzzy Rule-Based Untuk Otomatisasi Pintu Air dan PendeteksianEndapan,” *Jurnal Komputer Terapan*, vol. 8, no. 1, pp. 1–11, 2022, [Online]. Available: https://jurnal.pcr.ac.id/index.php/jkt/

[29] C. Asiminidis, G. Kokkonis, and S. Kontogiannis, “Database Systems Performance Evaluation for IoT Applications,” *International Journal of Database Management Systems*, vol. 10, no. 06, pp. 01–14, Dec. 2018, doi: 10.5121/ijdms.2018.10601.