 ISSN:2302-8149
e-ISSN:2540-9719

Architectural Design of Referral Patient Data Security using Advanced Encryption Standard

[1] Moh. Ali Romli *, [2]Muhammad Zakariyah
[1]Informatics, Faculty of Science & Technology, Universitas Teknologi Yogyakarta
[2]Medical Informatics, Faculty of Science & Technology, Universitas Teknologi Yogyakarta
[1,2]Sleman, Special Region of Yogyakarta, Indonesia
*e-mail: ali.romli@uty.ac.id

(***received***: 30 November 2024, ***revised***: 23 February 2025, ***accepted***: 25 February 2025)

Abstract

Electronic medical record can manage various kinds of patient data in digital form. Patient data security is a priority that must be met by healthcare provider for referral process. One of the medical data exchange standards that is widely used is Health Level Seven (HL7) standard. The absence of security in the HL7 standard makes patient data vulnerable to digital attacks, information security disturbances, and can even disrupt the patient's own psyche. This study aims to create an architectural design as well as a prototype of a patient data security system that uses HL7 standard, by utilizing the Advanced Encryption Standard (AES) as a cryptographic algorithm. Architectural design for data exchange, can change HL7 data from plain text and unauthenticated data transmission to data with secure and protected protocols. The research method starts from requirements analysis and finished with making system prototypes and model evolution. The system that has been developed is deployed into a SaaS model on cloud computing. The SaaS architecture for securing patient referral data has been adapted to the stakeholders involved (users), the medical data exchange standard used (HL7 standard), workflow and data exchange processes, and the data security technique itself (AES).

Keywords: referral patient, data security, electronic medical record, HL7, AES.

1 Introduction

Electronic Medical Record (EMR) is an electronic information about a person's health condition, which is created and managed together with doctors and authorized staff in a healthcare provider. EMR can facilitate workflow, while improving the quality of patient care and safety [1]. The EMR system can provide electronic copies of patient health information upon request. If the patient chooses to change healthcare provider, detailed medical records will help provide the healthcare provider destination with a comprehensive understanding.

In general, the use of EMR system to integrate health information is helpful for patients and clinicians in carrying out health care. However, when faced with data security risks, many patients are still concerned about the potential for the dissemination of personal information via the internet [2]. Data privacy in health care is very important for several reasons. Keeping patient information safe and confidential will help build trust from patients in the healthcare system. Electronic data security has been regulated by the previous government in UU ITE no. 11 of 2008 poin 30 and 32, while the medical record is regulated in Permenkes No. 269 of 2008 poin 10 and 12. The EMR regulation is still limited to the legal aspect of the law and has not explicitly regulated the privacy of EMR data itself, so that the development of RME at this time is still limited to replacing EMR from paper into electronic [3].

Health Level Seven (HL7) is one of the standards/protocols in the exchange, integration, sharing, and retrieval of health information electronically. This standard specifies how information is transmitted without system constraints, from one healthcare provider to another. HL7 supports interoperability of clinical practice, management, and evaluation of health care services, and the most used standard [4].

Over the years there have been several forms and improvements of HL7, but almost all versions of HL7 have one problem in common, namely security issues. No encryption or special checks are used, considering that this standard only regulates the language, structure, and data types needed for integration between systems [5]. Data security is crucial for healthcare provider and patients. The

 ISSN:2302-8149
e-ISSN:2540-9719

absence of security aspect in HL7 standard makes patient data vulnerable to digital attacks, information security disturbances, and can even disrupt the patient's own psyche. To reduce the possibility of digital attacks and penetration of information security using the HL7 standard, healthcare provider must focus on ensuring that information must be kept confidential, integrity, and available.

Applying data security techniques (commonly referred to cryptography) is the right choice to protect health data. Cryptography focuses on securing data directly, where health information is exchanged over the internet. Advanced Encryption Standard (AES) is the fastest encryption method that has flexibility, scalability, and is easy to implement [6], [7]. The AES algorithm has a high level of security because it uses up to 256-bit keys and resistant to various attacks such as square attacks, key attacks, key recovery attacks, and differential attacks. High performance with minimal storage space makes AES superior to other symmetric algorithms, which have many drawbacks regarding their performance and storage space.

This research aims to address a critical gap in HL7 security, especially in terms of weak encryption by designing and prototyping a secure patient data exchange system that leverages the HL7 standard enhanced with AES encryption. Specifically, this research will develop an architectural design for a secure HL7-based system that combines AES encryption and authentication mechanisms, implement a secure system prototype, and evaluate the prototype's effectiveness in ensuring the confidentiality and integrity of patient data during transmission.

The primary benefit of this research is to demonstrate the feasibility and effectiveness of securing HL7 data exchange using AES encryption. This will contribute to increasing patient trust in EMR systems [8], encouraging wider adoption of EMRs [9], and ultimately improving the quality of healthcare delivery [10]. This research is significant because it directly addresses a critical security vulnerability in current HL7 implementations that is hindering the full potential of EMR systems. By providing a practical and secure solution for patient data exchange, this work can pave the way for more resilient and trustworthy healthcare information systems. Existing research has focused primarily on the development and implementation of HL7, with little emphasis on addressing its inherent security weaknesses [11], [12]. This research fills this gap by focusing specifically on enhancing the security of HL7 through cryptographic means. Furthermore, this research will provide a practical demonstration of the effectiveness of AES encryption in a real-world healthcare context.

2 Literature Review

2.1 Previous Study

The importance of Electronic Medical Records (EMR) in modern healthcare has prompted extensive research in their development and implementation. Several studies have focused on improving the interoperability of EMR systems, with HL7 emerging as the dominant standard [13]. While these efforts have been successful in facilitating the exchange of data between disparate systems, HL7 security vulnerabilities have become a growing concern. Researchers have explored a variety of approaches to address security in healthcare information systems. Several studies have investigated the use of access control mechanisms to restrict unauthorized access to patient data [14]. Others have focused on implementing secure data storage solutions to protect against data breaches [15], [16]. However, the specific challenges of securing HL7 data transmission have received relatively less attention.

Several studies have touched on the security aspects of HL7, primarily focusing on the use of digital signatures for message authentication [17]. While digital signatures can provide integrity and non-repudiation, they do not address the fundamental issue of data confidentiality during transmission. Furthermore, the computational burden of digital signatures can be a concern in resource-constrained environments.

The use of cryptography, particularly AES, has been widely studied in the context of securing various types of data [18], [19]. However, its application to securing HL7 data transmission is still an area that requires further investigation. Some early work has explored the use of encryption in healthcare data exchange [20], [21], but a comprehensive and practical solution for securing HL7 using AES has not been fully developed.

 ISSN:2302-8149
e-ISSN:2540-9719

This study aims to address this gap by developing a robust and efficient mechanism for securing HL7 data transmission using AES encryption. Unlike previous studies that have focused primarily on other aspects of HL7 or used less robust security measures, this work specifically targets the confidentiality and integrity of HL7 data during transmission, leveraging the strengths of AES encryption. By providing a practical and secure solution, this study will contribute to the broader effort to build trustworthy and interoperable healthcare information systems.

2.2 Health Level Seven Standard

Health Level Seven (HL7) is an international community of health information experts who collaborate with each other to develop standards for health information exchange and health system interoperability. HL7 provides an international standard for how health data is stored, transferred, and shared among healthcare providers. This standard defines how data is packaged and transported including defining the languages, data types, and data structures required for inter-system integration.

HL7 helps bridge the gap in health information technology applications and makes it easier to share data and reduce administrative burdens in improving health services [22]. HL7 v2 is a pure messaging standard that allows the exchange of clinical data between systems to support patient care systems within a department. HL7 v2 messages consist of text, pipes (|), and caps (^). Messages and electronic documents are expressed as shown in Figure 1.

```
MSH|^~\&|NES|NINTENDO|TESTSYSTEM|TESTFACILITY|20010101000000||ADT^A04|Q12
3456789T123456789X123456|P|2.3
EVN|A04|20010101000000|||^KOOPA^BOWSER
PID|1||123456789^^^^EI|0123456789^AA^^JP|BROS^MARIO^LUIGI^II^DR^^L||19850101|M||
|123 FAKE STREET^MARIO \T\ LUIGI BROS PLACE^TOADSTOOL
KINGDOM^NES^A1B2C3^JP^HOME^^1234|1234|(555)555-0123^HOME^JP:1234567^^^^55
55550123||^ENGLISH|S|MSH|12345678||||||Y|3||||20180101090500|Y|||||HUMAN||PLUMBER
NK1|1|PEACH^PRINCESS|SO|ANOTHER CASTLE^^TOADSTOOL
KINGDOM^NES^^JP|(123)555-1234|(123)555-2345|NOK
NK1|2|TOADSTOOL^PRINCESS|SO|YET ANOTHER CASTLE^^TOADSTOOL
KINGDOM^NES^^JP|(123)555-3456|(123)555-4567|EMC
PV1|1|O|ABCD^EFGH||||123456^DINO^YOSHI^^^^^^MSRM^CURRENT^^^NEIGHBOURHO
OD DR
NBR|^DOG^DUCKHUNT^^^^^^^CURRENT||CRD|||||||123456^DINO^YOSHI^^^^^^MSRM^C
URRENT^^^NEIGHBOURHOOD DR
NBR|AO|0123456789|1|||||||||||||||||||MSH||A|||20010101000000
IN1|1|PAR^PARENT|123456789^^^^EI|||LUIGI|||||||||||||||||||||||||||||FAKE
IN1|2|FRI^FRIEND|123456789^^^^EI|||PRINCESS|||||||||||||||||||||||||||FAKE
```

Figure 1. HL7-v2 Example message
(Source: *https://blogs.mulesoft.com*)

3 Research Method

3.1 Advanced Encryption Standard Technique

Advanced Encryption Standard (AES) algorithm is a symmetric block cipher algorithm that takes plain text in 128 bit blocks and converts it into ciphertext using 128, 192, and 256 bit keys. The AES algorithm uses a substitution-permutation, with multiple loops to generate the ciphertext. The number of turns depends on the key used. A 128-bit key requires ten rounds, a 192-bit key requires 12 rounds, and a 256-bit key requires 14 rounds [23]. The schematic structure of AES is shown in Figure 2.

 ISSN:2302-8149
e-ISSN:2540-9719

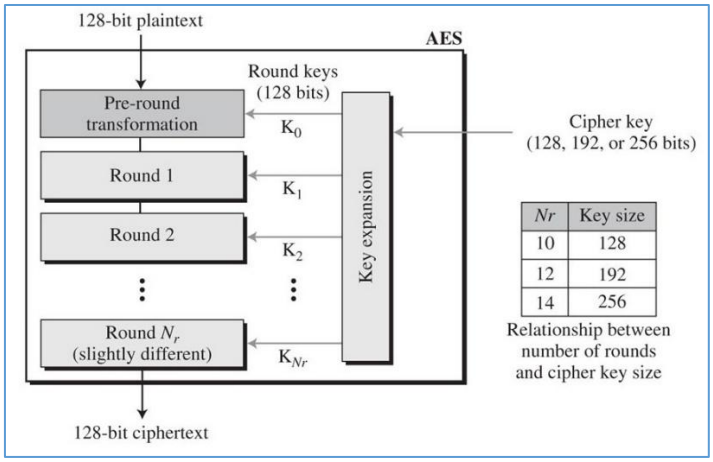

Figure 2. AES structure

(Source: www.blog.malwarebytes.com)

The encryption process in AES algorithm is carried out through 4 stages which are carried out repeatedly. These stages are:

1. Bytes substitution, block text bytes are replaced according to the rules defined by the predefined S-box (short for substitution box).
2. Shift rows, all rows except the first are shifted one block.
3. Mixing columns, randomize messages more by mixing column blocks.
4. Added round keys, XOR the messages with their respective round keys.

When performed repeatedly (according to the key used), these steps ensure that the final ciphertext is secure. The AES ciphertext decryption process is like the encryption process in reverse order.

3.2 Research Workflow

The research flow as presented in Figure 3, consists of 6 phases. The study begins by analyzing functional requirements (Phase I). At this phase, detailed analysis of the need for information systems in the healthcare environment is carried out. Workflows and data that will be used for communication between different healthcare provider are examined in more detail, specifically focusing on the types of data that need to be exchanged securely using HL7. This includes defining the specific HL7 message types required for identified workflows.

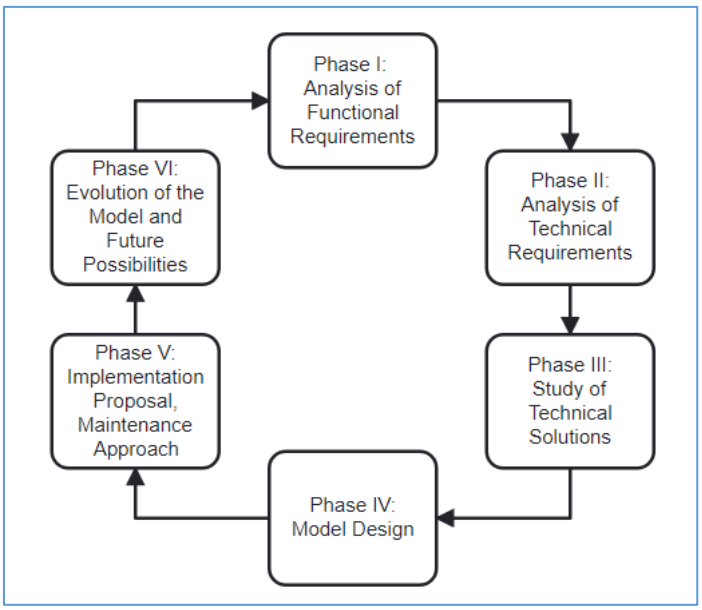

Figure 3. Research Workflow

The next phase is the process of conducting a technical requirements analysis (Phase II). The process of identifying technical requirements is carried out to determine a feasible platform to be implemented in a real environment. This process also identifies the hardware and software

 ISSN:2302-8149
e-ISSN:2540-9719

requirements that will be used for deployment needs. Most importantly, this phase also defines how AES encryption will be integrated with the chosen platform and HL7 messaging. This includes key management strategies, encryption/decryption processes, and performance considerations related to AES.

Phase III is the stage for evaluating the information exchange mechanism when carrying out the integration process. Based on the analyzed mechanism, an in-depth comparative study was conducted. This phase analyze how this mechanism can support secure HL7 communications using AES encryption. This involves comparing different approaches to incorporating AES into the HL7 message exchange process.

Model design (Phase IV) is a process for modeling the system according to the integration platform that includes the previously proposed functional and technical requirements, using the technical solutions analyzed in phase III. This phase details how AES encryption is applied to these messages, and how encrypted messages are sent and received. This design will define the use of AES to protect patient data within the HL7 framework.

Phase V is the proposed implementation and maintenance stage. The implementation of the integration model is proposed and considered, for the operation and maintenance of the platform. This phase focuses on developing a prototype that demonstrates secure HL7 message exchange using AES encryption. This includes implementing a key management system and encryption/decryption processes.

The last phase (Phase VI) is the evolution of the model and the possibility of future development. This stage is a process for possible future improvements and development of the platform, considering the expected evolution of the technology market in the healthcare sector, as well as the estimated future functional requirements. This phase also evaluate potential system enhancements, such as incorporating new versions of HL7 or exploring more sophisticated cryptographic techniques beyond AES, as necessary.

4 Results and Analysis

4.1 Functional Requirement

Analysis of functional requirements consists of input requirements, process, and output requirements. Input needs are data to be processed, while output needs are in the form of information that can later be retrieved and concluded. Analysis of functional requirements for SaaS services is shown in Table 1.

 ISSN:2302-8149
e-ISSN:2540-9719

Table 1. Functional Requirements Analysis

No	Process	Purpose	Input	Output
1.	User Registration	Create new users in the form of healthcare provider (hospital, health centers, clinical laboratories, and other health facilities).	The user registration process requires data such as username, password, name of healthcare provider, email, telephone number, and etc.	User account information
2.	User Log In/ Log Out	As a limitation of user access to SaaS system features.	Logging in need the username and password that were registered during the registration process.	Log In/Log Out Notification
3.	Create Referral Patient Data	A feature to secure (encrypt) patient data to be referred to the destination healthcare provider.	The input requirements in the referral process are Patient Name, Family Identification Number (NIK), and Medical Record File in HL7 v2 format with .txt extension.	Token
4.	Receive Referral Patient Data	A feature to open an encrypted reference file (decryption), so that it can be read by the destination healthcare provider.	The process of receiving a referral, requires a token that is generated by the origin healthcare provider through the creating referral process.	HL7 Files with .txt extension

Users involved in this system consist of super admin (government/stakeholders related to data security) and admin (hospitals, health centers, clinical laboratories, and other healthcare provider). In order to limit access, as well as limitation for system users, the admin registration process can only be carried out by super admin.

4.2 Proposed System Architecture

The main need for the system is how to secure referral patient data using the HL7 standard among several healthcare provider. The patient referral data security system uses SaaS services, designed to encrypt, and decrypt data using the AES algorithm by online. The system management is carried out centrally by third parties (government/stakeholders related to data security). Service users do not need to update or procure hardware or software, so it is expected to be more effective and efficient. The design of the system (SaaS system) to be implemented as shown in Figure 4.

The process of securing patient referral data is carried out through several stages. The first stage, original healthcare provider (Hospital 1) must register an account on the system. After registering and having an account, Hospital 1 then logs in and uploads a file containing patient referral data to be submitted to the destination healthcare provider (Hospital N). The file that is uploaded for encryption is the patient's medical record data in HL7 standard form. The encrypted file is stored in the system.

 ISSN:2302-8149
e-ISSN:2540-9719

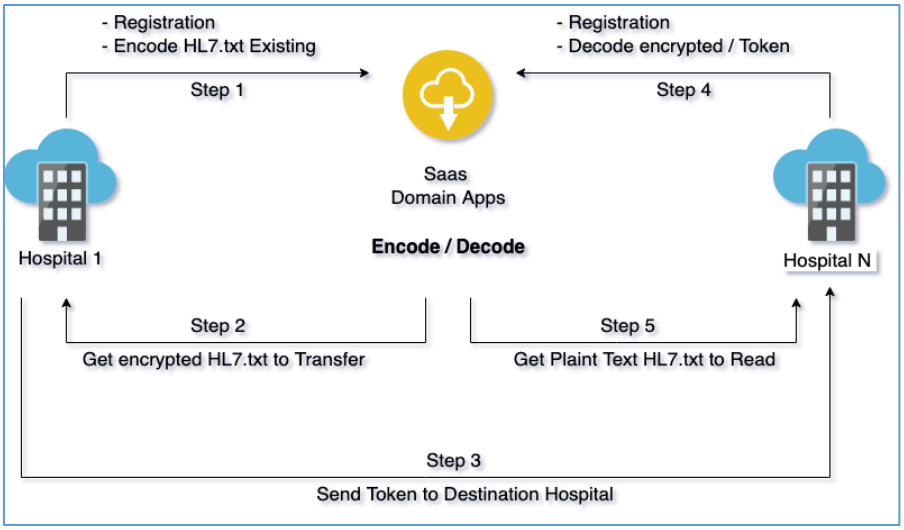

Figure 4. System architecture for referral patient data security using AES

After uploading the HL7 data, the system will generate tokens and Hospital N (who has received the token code and file from the referring Hospital 1) is tasked with carrying out the decryption process through the token code that has been received. Hospital N must also be registered in the system and log in using its account. Hospital N enters the token code to perform the decryption process of the HL7 file that has been encrypted. After the decryption process is carried out, Hospital N can download patient medical record data in the form of an HL7 file, according to the software available at Hospital N.

4.3 Implementation and Deployment

The software development infrastructure for this system includes a cloud computing platform. The infrastructure used for the implementation of the system is shown in Figure 5. This system can be accessed directly by the user via web browser, which will be bridged and connected to the Public IP address by CloudFlare. CloudFlare acts as a liaison between the system server and the user. The developed system is then integrated using one of the services in cloud computing, namely Software as A Service (SaaS). SaaS serves to store program code and HL7 encrypted files. SaaS is used as storage and database on this system.

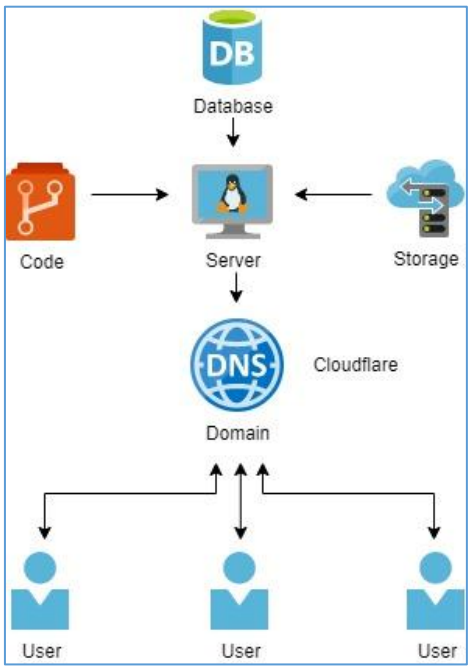

Figure 5. Software Architecture

 ISSN:2302-8149
e-ISSN:2540-9719

The components of the system development infrastructure used to develop, test, deploy, and manage software applications are shown in Table 2. The most common components of a software development infrastructure include source control management systems, build servers, test frameworks, and deployments. All components continuously work together to provide an efficient way to develop software applications.

Table 2. Software infrastructure components

No	Component	Description	Tools
1.	Integration Development Environment (IDE)	A software application that provides a Graphical User Interface (GUI) for managing the code base and building applications.	Visual Studio Code, JetBrains IntelliJ
2.	Workflow Management	Helping developers track and manage bugs and issues.	JIRA, Asana, Trello
3.	Source Control Management	Used to store and track changes to the codebase.	Git, GitHub
4.	Application Performance Management	Help developers to monitor and optimize the performance of their applications.	New Relics, AppDynamics
5.	Continous Integration and Deployment	To detect code changes, which are then used to generate system deployment artifacts. The tools also used to provide a mechanism to automatically apply code changes to staging environment, even to production environment.	GitLab, Jenkins
6.	Containerization Tools	Helping developers package their applications and dependencies into standalone units.	Docker, Kubernetes

The infrastructure components for software development as shown in Table 2 are components for developing the basic prototype of this SaaS system. A more complete infrastructure may be added as needed. The prototype dashboard that has been built on the SaaS service system for referral patient data security is shown in Figure 6 below.

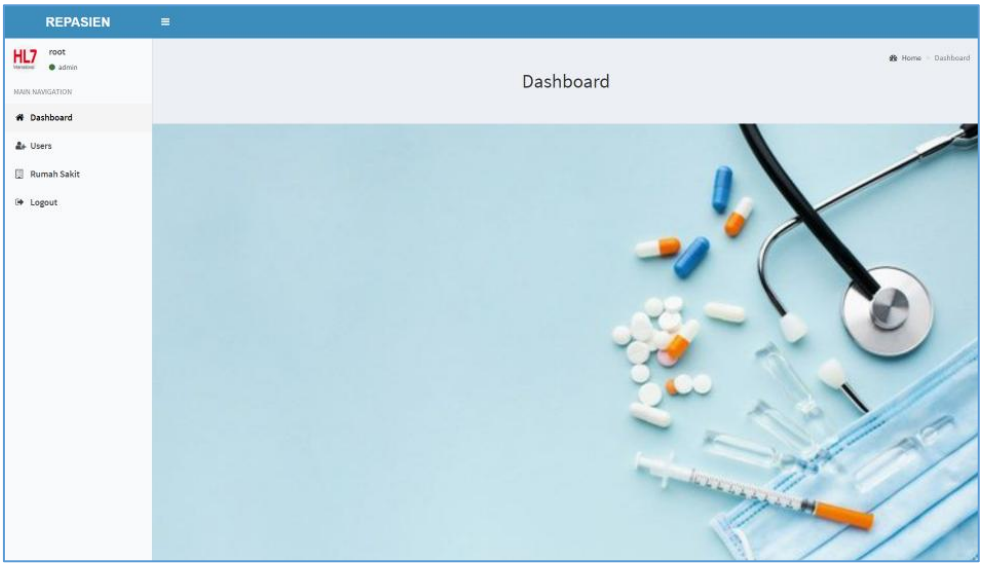

Figure 6. System prototype

4.4 System Testing

Unit testing is carried out to determine the reaction of the system to the unit being tested. This test is carried out by the admin and user, according to the unit being tested. Table 3 shows the results of unit testing on the referral patient data security system with the HL7 standard. In addition to functional testing of each feature owned by the system, testing of the system must also be carried out

 ISSN:2302-8149
e-ISSN:2540-9719

to ensure that the SaaS system is able to interact with users through notifications, confirmations, and other error handling. Figure 7 shows the form of error handling that may be performed by a user when using a SaaS system.

Table 3. Unit testing for the system

Unit	System Testing	Sistem Responses	Result	Tester
Login	Entering incorrect username and password.	Displays a message that the username and password are incorrect.	Success	Admin, Super Admin
	Enter username and password as super admin.	Displays the super admin dashboard page.	Success	Super Admin
	Enter username and password as admin.	Displaying the admin dashboard page.	Success	Admin
Create Referral Patient Data	Looking for patient referral history data based on the patient's name or NIK.	Displays patient referral history data based on the patient's name or NIK.	Success	Admin
	Add a referral by not entering the patient's name, patient NIK, or adding files.	Displays an error message according to the error made by the admin.	Success	Admin
	Add a referral by entering the patient's name, patient NIK, and adding the appropriate file.	Displays a message that performed process was successfully entered into the SaaS system database.	Success	Admin
Received Referral Patient Data	Enter the token code for the HL7 file decryption process.	Displays text to download HL7 files that can be downloaded by the admin.	Success	Admin

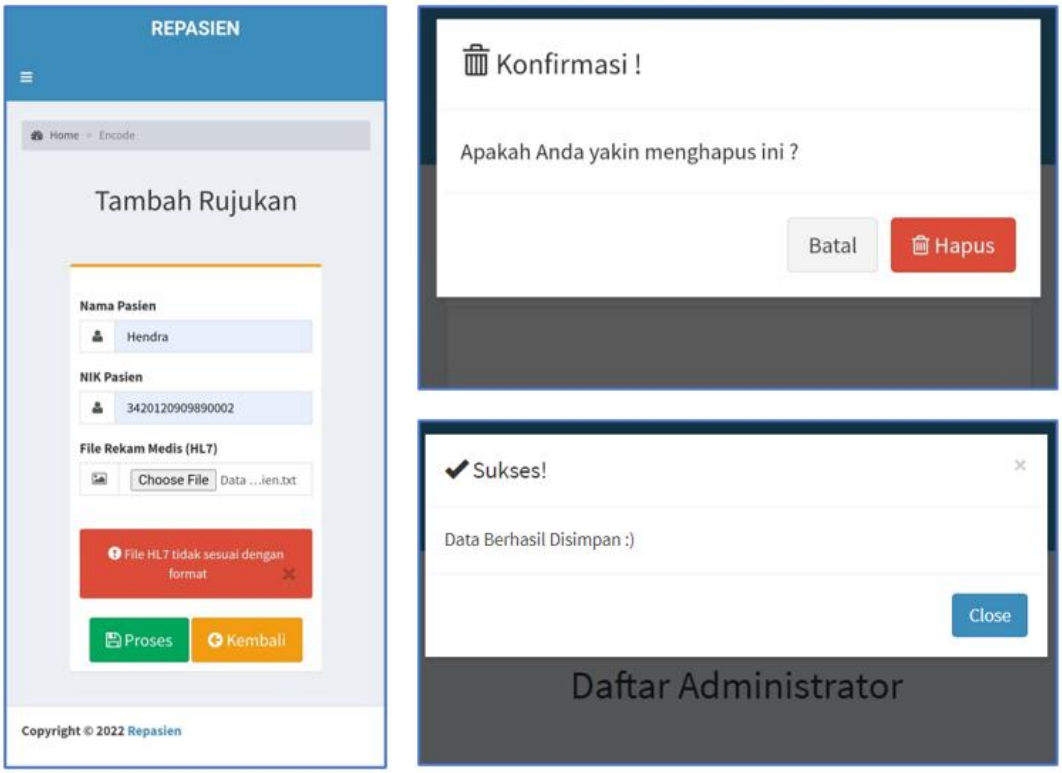

Figure 7. System testing for error handling

 ISSN:2302-8149
e-ISSN:2540-9719

4.5 Discussion

A well-designed software architecture can increase efficiency, reduce costs, improve communication between teams with different departments, and improve quality control [24], [25]. A clear and concise framework allows an organization to develop applications more efficiently, while ensuring that existing applications are compatible. As a result, a well-designed software development architecture can be an asset in any field.

The highest costs in software development are generally system maintenance and the addition of new features. However, if done early, evaluation of software architecture can reduce these costs by revealing the implications of architectural design [25]. The architecture makes expected evolution explicit and early error detection possible, reducing repair time during development as well as during operation.

Software architecture design for referral patient data security, requires infrastructure to support its performance. One of the things that need to be considered in terms of infrastructure development is deciding what programming language to use. The productivity of new software development seems to be influenced by the programming language used, while the productivity of developing existing software (adding features) seems to be less dependent on the programming language [26]. The programming language used will determine what tools and frameworks are needed to support the development process. After determining the programming language to use, the creation of a new development environment begins to emerge through options for code management, automated testing, continuous integration, and continuous deployment.

In addition to the technical aspects for creating a development environment, another thing that needs to be considered is infrastructure management related to the coordination of information technology resources, systems, platforms, and human resources used. Some of the most common types of technology infrastructure management include operating system management, cloud, virtualization, Information Technology (IT) operations, IT automation, container orchestration, configuration management, API management, risk management, and data management. However, the architecture of the SaaS system for securing patient referral data has been adjusted to the stakeholders involved, the medical data exchange standards used, the workflow and data exchange process, to the data security technique itself.

The software architecture for patient data security is SaaS-based, so the development and deployment process of the development infrastructure is in the cloud. Software as A Service (SaaS) is software that can be used and accessed without having to have a physical hardware system, because SaaS runs on a cloud-based server. SaaS can be used and accessed only with a stable internet connection. The advantage of using SaaS is that there is no need for expensive fees to buy physical server licenses and can save on maintenance and maintenance costs, because these things are already borne by the SaaS provider [27]. SaaS allow developers to concentrate on code development, rather than updating infrastructure. In addition, SaaS also provide increased security by controlling the most vulnerable parts of the development process by moving them from each developer's device to a central point [28].

Software development will continue to evolve along with the needs and technological developments. Developers need to be able to work with more complex data sets and build more user-friendly applications. Likewise with software for patient data security, it needs to be developed so that it can provide an API that may be collaborated with existing software in each healthcare provider. In addition, it is necessary to manage multiple projects and collaborate with other developers more effectively. Software for referral patient data security has performed several automations of repetitive tasks, reduced project cycle times, and improved communication and collaboration. Further development is expected to produce better quality software, especially to support administrative processes in healthcare provider in Indonesia.

5 Conclusion

Successful software development can be seen from careful planning, especially related to the design of the software architecture itself. To achieve this goal, it is very important to have a well-organized infrastructure. The process of defining the infrastructure must be clear, starting from the analysis of functional requirements to the analysis of components related to the software infrastructure

 ISSN:2302-8149
e-ISSN:2540-9719

itself. These processes have been defined in the creation of a SaaS system architecture for patient referral data security. The architecture has been adapted to the stakeholders involved, the medical data exchange standards used, the workflow and data exchange processes, to the data security techniques themselves.

Acknowledgement

This work would not have been possible without the financial support of the Ministry of Education, Culture, Research, and Technology through Penelitian Dosen Pemula Scheme. I am grateful to all of those with whom I have had the pleasure to work on this and other related projects. Each of the members of my research program has provided us with extensive personal and professional guidance and taught me a great deal about both scientific research and life in general.

References

[1] S. Upadhyay and H. Hu, "A Qualitative Analysis of the Impact of Electronic Health Records (EHR) on Healthcare Quality and Safety: Clinicians' Lived Experiences," *Health Serv Insights*, vol. 15, Jan. 2022, doi: 10.1177/11786329211070722.

[2] M. Duckert and L. Barkhuus, "Protecting Personal Health Data through Privacy Awareness," *Proc ACM Hum Comput Interact*, vol. 6, no. GROUP, pp. 1–22, Jan. 2022, doi: 10.1145/3492830.

[3] N. Amalia, M. Z. A. Rustam, A. Rosarini, D. R. Wijayanti, and M. A. Riestiyowati, "The Implementation of Electronic Medical Record (EMR) in The Development Health Care System in Indonesia," *International Journal of Advancement in Life Sciences Research*, vol. 4, no. 3, Jul. 2021, doi: 10.31632/ijalsr.2021.v04i03.002.

[4] R. Ait Abdelouahid, O. Debauche, S. Mahmoudi, and A. Marzak, "Literature Review: Clinical Data Interoperability Models," *Information*, vol. 14, no. 7, p. 364, Jun. 2023, doi: 10.3390/info14070364.

[5] C. Thapa and S. Camtepe, "Precision health data: Requirements, Challenges and Existing Techniques for Data Security and Privacy," *Comput Biol Med*, vol. 129, p. 104130, Feb. 2021, doi: 10.1016/j.compbiomed.2020.104130.

[6] A. Olutola and M. Olumuyiwa, "Comparative Analysis of Encryption Algorithms," *European Journal of Technology*, vol. 7, no. 1, pp. 1–9, Jan. 2023, doi: 10.47672/ejt.1312.

[7] X. Guo, M. El-Hadedy, S. Mosanu, X. Wei, K. Skadron, and M. R. Stan, "Agile-AES: Implementation of Configurable AES Primitive with Agile Design Approach," *Integration*, vol. 85, pp. 87–96, Jul. 2022, doi: 10.1016/j.vlsi.2022.04.005.

[8] O. Enaizan, B. Eneizan, M. Almaaitah, A. T. Al-Radaideh, and A. M. Saleh, "Effects of Privacy and Security on the Acceptance and usage of EMR: The Mediating Role of Trust on the basis of Multiple Perspectives," *Inform Med Unlocked*, vol. 21, p. 100450, 2020, doi: 10.1016/j.imu.2020.100450.

[9] A. F. Dennis, P. J. White, and T. Zayas-Cabán, "Fast-Tracking Health Data Standards Development and Adoption in Real-World Settings: A Pilot Approach," *Appl Clin Inform*, vol. 12, no. 04, pp. 745–756, Aug. 2021, doi: 10.1055/s-0041-1731677.

[10] M. Anywar *et al.*, "Challenges and Lessons Learned in Mapping HL7 v2 Data to openEHR: Insights from UKSH Medical Data Integration Center," 2024. doi: 10.3233/SHTI240658.

[11] G. Dupont, D. R. dos Santos, E. Costante, J. den Hartog, and S. Etalle, "A Matter of Life and Death: Analyzing the Security of Healthcare Networks," 2020, pp. 355–369. doi: 10.1007/978-3-030-58201-2_24.

[12] A. L. Martínez, M. G. Pérez, and A. Ruiz-Martínez, "A Comprehensive Model for Securing Sensitive Patient Data in a Clinical Scenario," *IEEE Access*, vol. 11, pp. 137083–137098, 2023, doi: 10.1109/ACCESS.2023.3338170.

[13] A. A. AlQudah, M. Al-Emran, and K. Shaalan, "Medical data integration using HL7 Standards for Patient's Early Identification," *PLoS One*, vol. 16, no. 12, p. e0262067, Dec. 2021, doi: 10.1371/journal.pone.0262067.

[14] M. Rizwan *et al.*, "Risk Monitoring Strategy for Confidentiality of Healthcare Information," *Computers and Electrical Engineering*, vol. 100, p. 107833, May 2022, doi: 10.1016/j.compeleceng.2022.107833.

 ISSN:2302-8149
e-ISSN:2540-9719

[15] B. Seth, S. Dalal, V. Jaglan, D. Le, S. Mohan, and G. Srivastava, "Integrating Encryption Techniques for Secure Data Storage in the Cloud," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 4, Apr. 2022, doi: 10.1002/ett.4108.

[16] A. Almalawi, A. I. Khan, F. Alsolami, Y. B. Abushark, and A. S. Alfakeeh, "Managing Security of Healthcare Data for a Modern Healthcare System," *Sensors*, vol. 23, no. 7, p. 3612, Mar. 2023, doi: 10.3390/s23073612.

[17] T. Gunasekar, P. D. D. Dominic, and S. Kayalvizhi, "Authentic Cloud-Biometric Signature Verification System for Healthcare Data Management," *Int J Bus Inf Syst*, vol. 37, no. 1, p. 63, 2021, doi: 10.1504/IJBIS.2021.115069.

[18] A. Kumar, "Data Security and Privacy using DNA Cryptography and AES Method in Cloud Computing," in *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, IEEE, Nov. 2021, pp. 1529–1535. doi: 10.1109/I-SMAC52330.2021.9640708.

[19] M. Kumar, A. Soni, A. R. S. Shekhawat, and A. Rawat, "Enhanced Digital Image and Text Data Security using Hybrid Model of LSB Steganography and AES Cryptography Technique," in *2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, IEEE, Feb. 2022, pp. 1453–1457. doi: 10.1109/ICAIS53314.2022.9742942.

[20] K. Shankar Komathi Maathavan and S. Venkatraman, "A Secure Encrypted Classified Electronic Healthcare Data for Public Cloud Environment," *Intelligent Automation & Soft Computing*, vol. 32, no. 2, pp. 765–779, 2022, doi: 10.32604/iasc.2022.022276.

[21] J. Jain and A. Jain, "Securing E-Healthcare Images using an Efficient Image Encryption Model," *Sci Program*, vol. 2022, pp. 1–11, Mar. 2022, doi: 10.1155/2022/6438331.

[22] J. Nan and L.-Q. Xu, "Designing Interoperable Health Care Services based on Fast Healthcare Interoperability Resources: Literature Review," *JMIR Med Inform*, vol. 11, p. e44842, Aug. 2023, doi: 10.2196/44842.

[23] S. M. Kareem and A. M. S. Rahma, "New Method for Improving Add Round Key in the Advanced Encryption Standard Algorithm," *Information Security Journal: A Global Perspective*, vol. 30, no. 6, pp. 371–383, Nov. 2021, doi: 10.1080/19393555.2020.1859654.

[24] S. Alsaqqa, S. Sawalha, and H. Abdel-Nabi, "Agile Software Development: Methodologies and Trends," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 14, no. 11, p. 246, Jul. 2020, doi: 10.3991/ijim.v14i11.13269.

[25] Z. Fang, "System-of-Systems Architecture Selection: A Survey of Issues, Methods, and Opportunities," *IEEE Syst J*, vol. 16, no. 3, pp. 4768–4779, Sep. 2022, doi: 10.1109/JSYST.2021.3119294.

[26] J. Gmys, T. Carneiro, N. Melab, E.-G. Talbi, and D. Tuyttens, "A Comparative Study of High-Productivity High-Performance Programming Languages for Parallel Metaheuristics," *Swarm Evol Comput*, vol. 57, p. 100720, Sep. 2020, doi: 10.1016/j.swevo.2020.100720.

[27] S. Raghavan R., J. K.R., and R. V. Nargundkar, "Impact of Software as a Service (SaaS) on Software Acquisition Process," *Journal of Business & Industrial Marketing*, vol. 35, no. 4, pp. 757–770, Apr. 2020, doi: 10.1108/JBIM-12-2018-0382.

[28] S. Galiveeti, L. Tawalbeh, M. Tawalbeh, and A. A. A. El-Latif, "Cybersecurity Analysis: Investigating the Data Integrity and Privacy in AWS and Azure Cloud Platforms," 2021, pp. 329–360. doi: 10.1007/978-3-030-74575-2_17.