Applied Information Technology And Computer Science Vol. 4 No. 1 (2023) 059-078

AITCS

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/aitcs
e-ISSN :2773-5141

Care Foundation Electronic Medical Record System Cryptographic-Based Approach and Role-Based Access Control

Khoo Xiao Hui[1], **Sofia Najwa Ramli***[1]

[1]Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author Designation

DOI: https://doi.org/10.30880/aitcs.2023.04.01.004
Received 21 July 2022 Accepted 18 June 2023; Available online 30 June 2023

Abstract: An electronic medical record (EMR) system is software that allows electronic entry, storage, and maintenance of digital medical data. EMR systems have been a great asset to the healthcare community. However, a significant number of privacy issues are raised by a shift from paper-based systems to integrated and electronic ones. Data breaches in the healthcare industry are increasing rapidly, and the general adoption rate of EMR is comparatively low due to security and privacy issues. Therefore, this project intends to develop a secure EMR system that integrates cryptography and role-based access control into the implementation of the system based on the Object-Oriented Software Development (OOSD) methodology using Jakarta Servlet. The user acceptance testing result shows that the EMR system is generally well-accepted by the end-users. Overall, all the objectives have been achieved. Future work may consider expanding the scope of development in terms of platform and functions.

Keywords: Electronic Medical Record, Cryptography, Role-Based Access Control, Jakarta Servlet, Object-Oriented Software Development (OOSD)

1. Introduction

According to the Center of Medicare and Medicaid Services (CMS), an electronic health record (EHR) is defined as a patient's medical history that is maintained electronically by the provider, which may include key administrative clinical data of the patient's care under a medical staff, such as progress notes, vital signs, medications, past medical history, and more. Recent studies have shown that integrated health records have many advantages, such as improved quality of care, cost reduction, increased efficiency, and more accurate patient information [2]. However, a study by Rothstein [1] stated that due to a significant number of privacy issues raised by a shift from paper-based systems to integrated and electronic ones, there has been limited activity in policy development. Most hospitals

*Corresponding author: sofianajwa@uthm.edu.my

around the world employ EMRs, but their acceptance among healthcare practitioners has yet to recognize the current issues in terms of privacy and security [3]. The healthcare industry is one of the top three industries experiencing the largest number of security breaches [4]. In the first half of 2020

alone, there were 225 healthcare data breach incidents, in which 130 of these breaches were caused by hacking/IT events, accounting for 57.77% of the total number of attacks [6], while 59 of them were caused by internal unauthorized access. In one example, critical security vulnerabilities in an open-source EMR system called nTreatment have led to thousands of patient records being exposed on the Internet [21]. According to a report by [20], sensitive health data and patient information were not encrypted and not protected with passwords.

Additionally, previous studies [5],[18] have reported that the general adoption rate of EMR is comparatively low. As of 2019, 25% of the 145 government hospitals in Malaysia used Hospital Information System (HIS), whereas only 7% of government clinics are equipped with Clinical Information System (CIS). Several clinical system implementations have failed because of the reluctance of EMR adoption by physicians [9]. The transition of the healthcare information system from a paper-based to an automated system raises concerns about safety, privacy, and ethics. These problems were revealed to be impeding physicians' acceptance and use of EMR [8].

Moreover, privacy, security, and confidentiality are the challenges that must be addressed in EMR systems. For example, cases of healthcare professionals disclosing famed individuals' health records, identity theft, or unintentional disclosure of health records through a stolen smartphone. In an article published in the HIPAA Journal [22], two employees of Vanderbilt University Medical Center had been inappropriately accessing the medical records of over 3000 patients. Medical records of patients continued to be accessed without permission for over a year. Preliminary studies [7] revealed that most patients want more control over the use of data kept in their EMR, and none consented to full and comprehensive access to their EMR by physicians associated with their healthcare provider. Based on these issues mentioned, the objectives identified for this project are:

1. To design a secure EMR system using the object-oriented approach.
2. To develop the proposed system with security features using cryptography and role-based approaches.
3. To test the performance of the proposed system in terms of functional and security requirements.

The EMR system focuses primarily on implementing security techniques like cryptography and role-based access control (RBAC). The target users of the EMR system are the admin, doctor, receptionist, and patient. The proposed system has six main modules: login, user registration, patient registration, book appointment, medical records, and audit log.

2. Related Work

This section analyses and discusses in detail the features of two existing open-source EMR systems in terms of functionality and security. The selected EMR systems are OpenEMR and ClearHealth.

2.1 Open Electronic Medical Record (OpenEMR)

OpenEMR is open-source PHP-based software that runs on the Apache webserver and the MySQL database server. It includes many important features for clinical practices, such as patient data feeding (e.g., biographic data, diagnostic results, medication history) EHR, disease management, scheduling, and electronic billing. OpenEMR comes with a set of pre-defined roles and permissions to which an administrator can assign users to. The system supports RBAC and the capability to revoke a user's permissions [12]. It also supports a Patient Portal that allows patients to make appointments.

In terms of security, a study [11] found that OpenEMR encrypts medical documents using the PHP mcrypt library. However, the encryption algorithm used is the TripleDES algorithm, which is known to be less secure than AES. Furthermore, another study [17] has identified some of the authentication flaws in OpenEMR 4.1.1. For instance, there is no forgotten password option to assist users in recovering their passwords if they forget them. Moreover, there is no account lockout after a set number of failed tries. This makes the system more susceptible to brute-force attacks. The proposed system locks the user out if three incorrect attempts have been made to prevent unauthorized access. The failed attempts are recorded in the log to ensure that the admin can be aware of the unusual activity.

2.2 ClearHealth

ClearHealth is a PHP-based open-source EMR system that features Electronic Medical Records (EMR), recording patient demographics, medical billing, medical accounts receivable, scheduling, and access control [10]. ClearHealth was built with HIPAA-compliant security measures. The PHPGacl (PHP General Access Control Lists) toolbox was used to implement the security features [16]. RBAC can be established using the toolkit. One of the security aspects is that the admin section is not always shown. The admin section allows users to adjust it so that it can be tailored to their office or practice's specific needs. There is also a timer that enables automatic logoff as a security measure [19]. This feature is implemented in the proposed system in compliance with the HIPAA Security Rule of Automatic Logoff Procedures.

However, when it comes to secure data storage, research [15] has shown that ClearHeath did not provide this security measure. The proposed system is able to securely store patients' personal information and medical records through encryption to protect the privacy of the patients. Moreover, the logging mechanism in ClearHealth is not identified as well [15]. When conflicts emerge over significant concerns such as authorization misuse, unlawful access attempts, and inappropriate disclosure of patients' health data, audit trails can serve as proof [14]. Therefore, audit trails are included in the proposed system to help admins monitor and track the activities of the users within the EMR system.

2.3 Comparison of Existing Systems with Care Foundation EMR system

In this section, the security features that are adopted against which the previously discussed EMR systems are compared, are selected based on HIPAA Security Rule and OWASP secure coding checklist. The security measures are data encryption, password hashing, audit log, authentication, automatic log-off, access control, account lockout policy, and password protection.

Table 1: Comparison of security features between the systems

Security features	OpenEMR	ClearHealth	Care Foundation
Data encryption	TripleDES	No	AES
Password hashing	No	No	Yes
Audit log	Yes	No	Yes
Authentication	Password-based	Password-based	Password-based
Automatic log-off	Yes	Yes	Yes
Access control	RBAC	RBAC	RBAC
Account lockout policy (after 3 attempts)	No	No	Yes
Password policy	Yes	No	Yes

Table 1 depicts the comparison results of the three systems. The most common security features implemented in the set of systems are authentication, automatic log-off, and access control. In contrast,

password hashing, and account lockout policy are totally absent from both existing open-source EMR systems. It appears that RBAC is used as the access control model in all systems. Out of the eight security measures, OpenEMR met six of the requirements, whereas ClearHealth only met three of them.

3. Methodology

Care Foundation EMR system is developed using the Object-Oriented Software Development (OOSD) model. A system is considered a collection of items in this approach. The key distinction between a traditional technique, such as structured design, and an object-oriented approach is the manner in which a problem is dissected. The problem-decomposition procedure in traditional methodologies is either process-centric or data-centric [13]. Therefore, the Unified Modelling Language (UML) diagramming approach is developed and incorporated into this methodology. The major phases of software development using an object-oriented approach are object-oriented analysis, object-oriented design, and object-oriented implementation. The following sections elaborates on the activities to be carried out in each phase of the OOSD model and their deliverables.

3.1 Planning Phase

In this phase, the project is initiated by proposing a project title and discussing it with the supervisor. Once a suitable title has been chosen, the next step is to determine the problem statement which is done by identifying security issues in existing EMR systems. The solutions that are proposed based on the identified problems are then implemented in the proposed EMR system. The objectives, scope, expected outcome, and project significance are also determined and documented in the project proposal. The output of this phase is a project proposal that includes the project plan and Gantt chart.

3.2 Analysis Phase

This is the second phase of the OOSD model that involves the process of collecting functional and security requirements for the proposed system. The behaviors and characteristics of the main users: admin, doctor, receptionist, and patient, are the elements that form the functional requirements for the proposed system. Since the aim of this project is to develop a secure EMR system, therefore the security aspect is focused on more compared to its functionality. The process of determining the security requirements is divided into three parts: document analysis, regulatory and compliance requirements identification, and architectural risk analysis. At the end of the Object-Oriented Analysis phase, the output would be the functional and security requirements for the proposed EMR system and abuse cases.

3.3 Design Phase

In this phase, the goal is to transform the list of requirements into designs that include detailed specifications covering all aspects of the EMR system. To accomplish this, UML (Unified Modeling Language) diagrams are used to define the application structure, system behavior, and business processes within the EMR system. The deliverables of the design phase include a class diagram, use case diagrams, sequence diagrams, activity diagrams, system architecture diagram, an ERD, and lastly the user interface designs.

3.4 Implementation Phase

Once the design is completed, the EMR system is developed based on the designs. The web application is based on Java using Servlets and JSP, which can be run using the Apache Tomcat server. MySQL database is used to store and retrieve information. Moreover, the hardware requirements for developing the proposed system include a processor with the 10th Generation Intel Core i5-1035G1 that has quad-core processors with 6MB of cache, 8GB of random-access memory (RAM) at a minimum, and a 475

GB hard drive for storage. By the end of this phase, there is a completely functional EMR system that fulfills the functional and security requirements in the analysis phase.

3.5 Testing Phase

The tests are performed by focusing on unit test cases to validate the software security and business logic. The result of each test is marked as pass or fail depending on the output of the system. A user acceptance test is also conducted using Google Form. Medical practitioners and receptionists are invited to participate in the survey test by interacting with the system and providing their feedback afterward. The deliverables of the testing phase are the test plan, test results, bugs found, and the user acceptance results.

4. System Analysis and Design

4.1 Functional Requirement Analysis

Care Foundation EMR system has six main modules: login, user registration, managing patient details, booking appointments, managing medical records, and audit log. Table 2 shows the functional modules of Care Foundation EMR system.

Table 2: The functional requirements for Care Foundation EMR system

Requirement ID	Functional Module	Description
FR001	Login	The system should require users to provide a valid username and password before accessing the system.
FR002	User registration	The system should allow the admin to register new users based on their roles.
FR003	Book appointment	The system should allow patients to book an appointment based on the availability and the schedule of the selected doctor by providing personal details.
FR004	Manage medical records	The system should let doctors view, insert, and update the medical record of the patients assigned to them.
FR005	Manage medical history	The system should let doctors view, insert, and update the medical history of the patients assigned to them.
FR006	Manage patient details	The system should allow receptionists to view, insert, and update the details of the patients that they registered.
FR007	Manage patient appointments	The system should let receptionists view, insert, and update appointment details and assign doctors to the patients that they registered.

4.2 Security Requirement Analysis

This section analyses the security requirements using the Abuse Case for identifying the attacks against EMR systems. The potential attackers of the EMR system are divided into two categories: a malicious insider, and an outsider hacker. The abuse cases for both attackers are depicted in Figure 1(a) and 1(b) respectively.

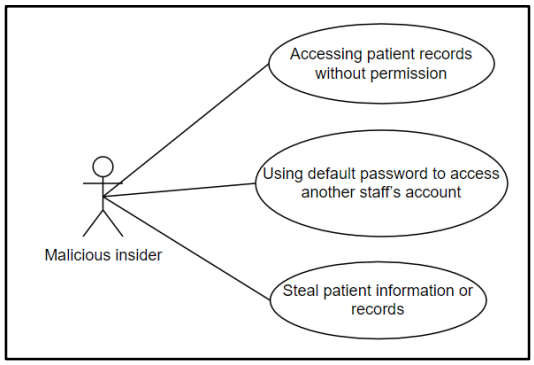


Figure 1(a): Abuse case scenario for malicious insider

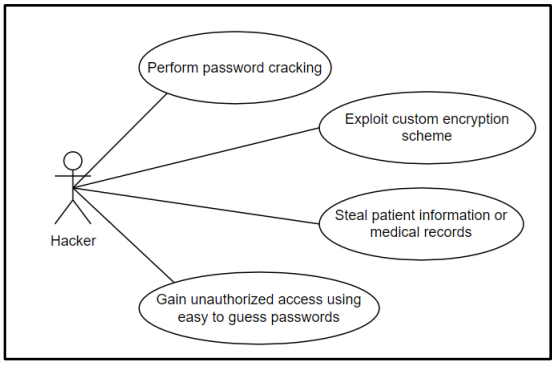


Figure 1(b): Abuse case scenario for a hacker

There have been cases where malicious employees accessed patient records without permission [22] and abused their access privileges to review and steal over 10000 patient data as part of a fraudulent driver's license scheme [23]. In addition, there was a report of a new wave of large-scale password spraying campaigns directed at healthcare with the likely goal of stealing information related to the coronavirus outbreak [24]. Another report by [25] wrote that an analysis by NordPass, a proprietary password manager, found that healthcare executives are increasingly exposing their companies to more breaches due to weak passwords.

Based on the analysis above and research performed, the security requirements for Care Foundation EMR system are summarized in Table 3.

Table 3: The security requirements for Care Foundation EMR system

Requirement ID	Security Module	Description
SR001	Change password	The system should require the staff to change their password when they log in for the first time.
SR002	Password policy	The system should require users to set strong passwords by ensuring that the passwords meet the complexity requirement.
SR003	Password hashing	The system should be able to hash the passwords in the database to provide authentication.
SR004	Authentication	The system should grant access only to users who provide both valid email and password during login.
SR005	Account lockout policy	The system should lock the account when three incorrect login attempts have been made within a set amount of period.
SR006	Access control	The system should implement RBAC to restrict EMR access to users based on their role.
SR007	Audit logging	The system should be able to record the user ID, username, event, date, time, and description of all user activities into the database.
SR008	Automatic log-off	The system should automatically log out a user after 10 minutes of inactivity.
SR009	Data encryption	The system should be able to implement a strong cryptographic algorithm to encrypt sensitive information in the database.

4.3 System Design

A use case diagram illustrates a model scenario in which the actors interact with a system using specific symbols and connections. The actors of the EMR system are the administrator, doctor, receptionist, and patient. Figure 2 illustrates the use case diagrams for each of the target users of the EMR system.

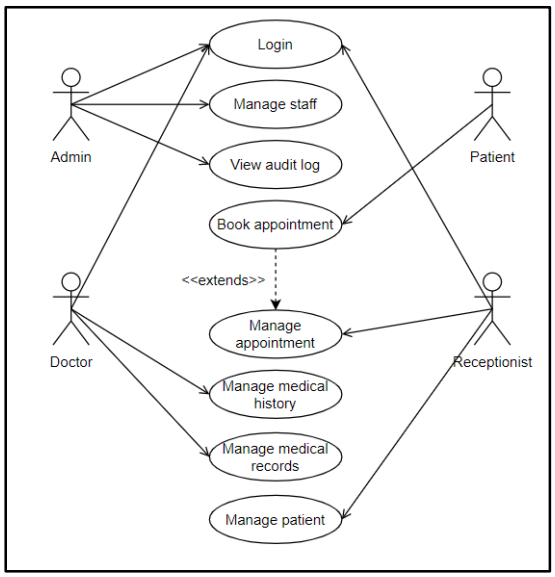


Figure 2: Use case diagram for the admin, doctor, receptionist, and patient

There are a total of eight classes in the class diagram as illustrated in Appendix A. The classes include Login, Audit Log, Staff, Key Server, Book Appointment, Manage Appointment, Medical Record, and Medical History.

An activity diagram is a flowchart that depicts the flow from one activity to another. The action can be defined as a system operation. The primary goal of activity diagrams is to illustrate the system's dynamic behavior. The activity diagrams for the admin, receptionist, doctor, and patient are shown in Figures 3(a), 3(b), 3(c), and 3(d) respectively.

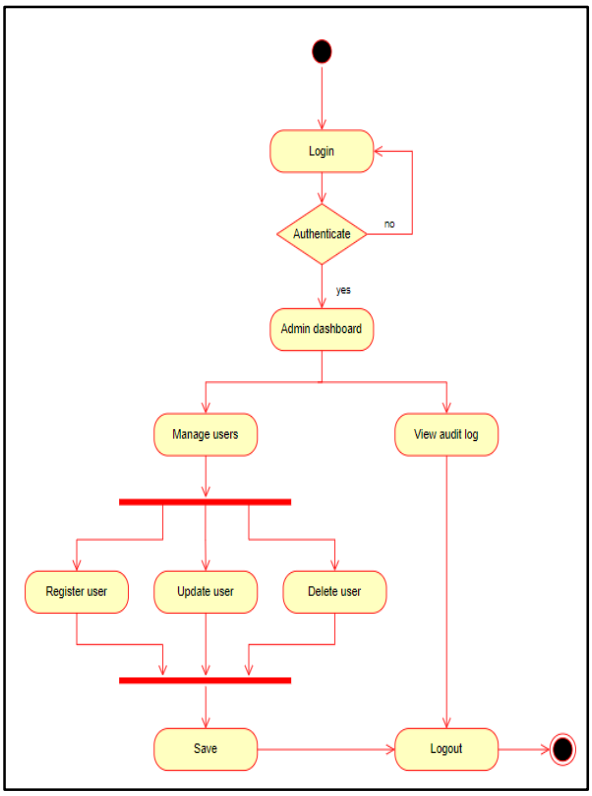


Figure 3(a): Activity diagram for admin

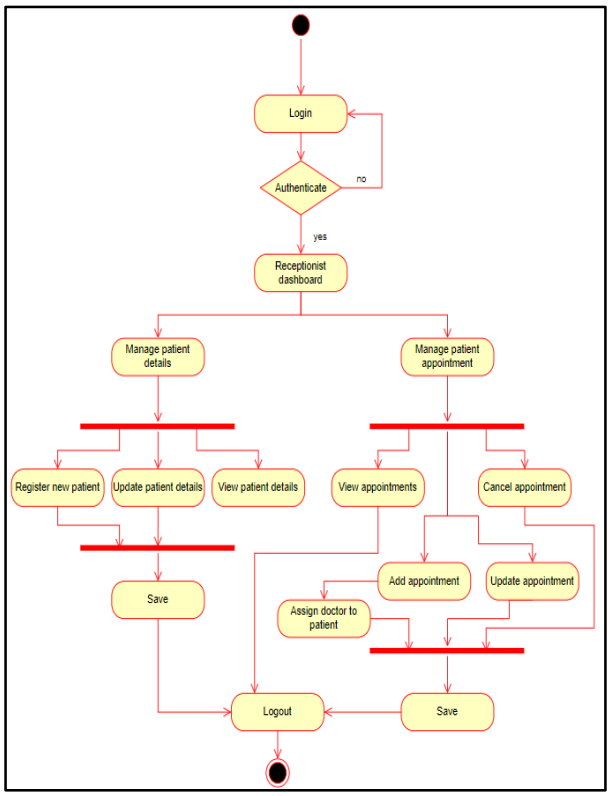


Figure 3(b): Activity diagram for receptionist

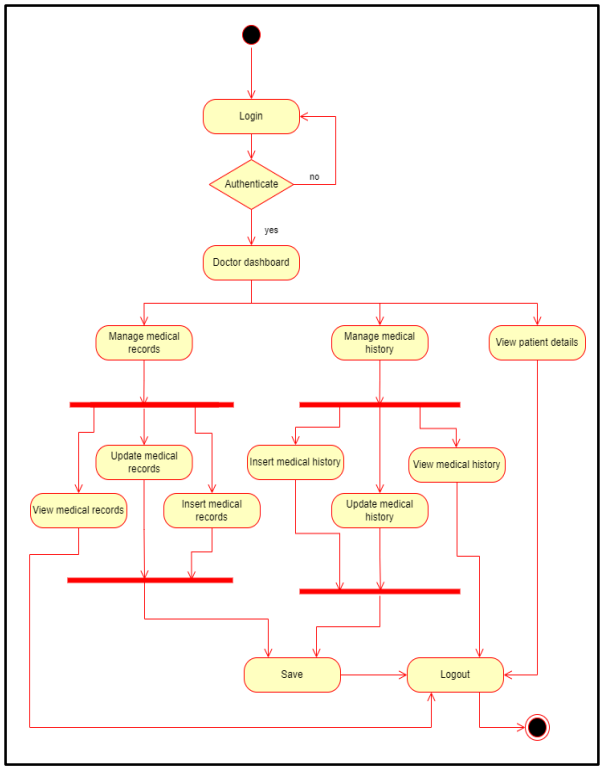


Figure 3(c): Activity diagram for doctor

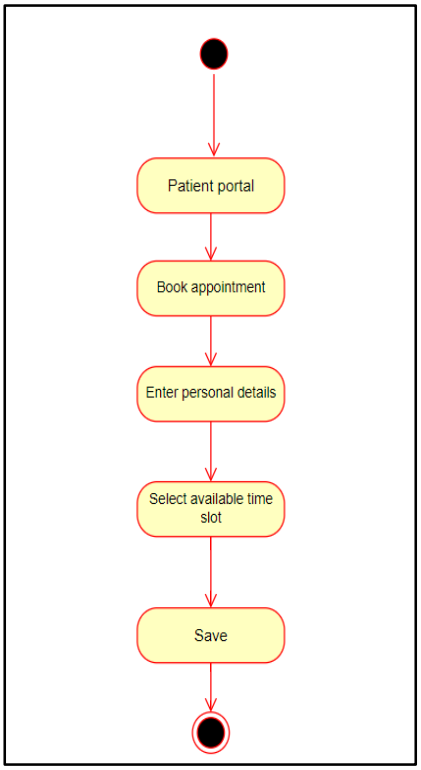


Figure 3(d): Activity diagram for patient

Since the EMR system stores a lot of sensitive information, the AES encryption and SHA512 hashing algorithm are used to secure the data at rest in the database. Table 4 gives an overview of where cryptography is applied to what data.

Table 4: An overview of the application of cryptography in Care Foundation EMR system

Module	Table Name	Data Protected	Crypto Algorithm
User registration, Login	Staff	Password	PBKDF2WithHmac SHA512
Manage patient appointments	Appointment	Patient name and contact number	AES algorithm
Manage medical records	Medical Record	History of present illness, examination, diagnosis, treatment, remark	AES algorithm
Manage medical history	Medical History	Medical history, allergies, family history, medication taken	AES algorithm
Manage patient details	Patient Info	Patient name, IC number, address, postcode, city, and contact number	AES algorithm

Care Foundation EMR system uses RBAC to assign permission to the staff based on their role within the hospital. For extra security, each staff member is only allowed to view the sensitive records on a need-to-know basis. In other words, the staff member can only view the records that are either created by them or assigned to them. Table 5 summarizes the implementation of RBAC in Care Foundation EMR system. Users with permission to access the table are marked with a check symbol.

Table 5: Summary of RBAC implementation in Care Foundation EMR system

Table Name	Admin	Receptionist	Doctor
Appointment		✓	
Audit log	✓		
Medical History			✓
Medical Record			✓
Patient Info		✓	✓
Staff	✓		

5. Implementation and Testing

5.1 Implementation of Security Module

This section highlights some of the significant security modules added to the proposed system by including code snippets for that module and its interface design. The security modules that are discussed are password hashing, audit logging, and data encryption. Figure 4 shows the code snippet for password hashing.

```
public static String getSalt(int length) {
    StringBuilder returnValue = new StringBuilder(length);
    for (int i = 0; i < length; i++) {
        returnValue.append(ALPHABET.charAt(RANDOM.nextInt(ALPHABET.length())));
    }
    return new String(returnValue);
}
public static byte[] hash(char[] password, byte[] salt) {
    PBEKeySpec spec = new PBEKeySpec(password, salt, ITERATIONS, KEY_LENGTH);
    Arrays.fill(password, Character.MIN_VALUE);
    try {
        SecretKeyFactory skf = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA512");
        return skf.generateSecret(spec).getEncoded();
    } catch (NoSuchAlgorithmException | InvalidKeySpecException e) {
        throw new AssertionError("Error while hashing a password: " + e.getMessage(), e);
    } finally {
        spec.clearPassword();
    }
}
```

Figure 4: Code snippet for generating salt value and computing a salted hashed password

First, a salt value is randomly generated using the getSalt() function. The password hashing is done by calling the hash() function, which takes two parameters: the input password and the salt value. The "PBKDF2WithHmacSHA512" produces a hash length of 512 bits, which is more secure compared to "PBKDF2WithHmacSHA1". Figure 5 shows the hashed passwords and salt being stored in the database.

staff_id	role_id	staff_name	password	salt
ST010	R002	Aaron Lee Kwang Yang	+KY2TZkkPApnKSo0nykj8EVmveuWYLsJG9N2jQqIhLE=	xf1ayEwTlHxuFlApDq1etftmqoq953
ST011	R002	Jeremy Tan Tian Hui	/HP4ZjJoLkwLQeQsgj2ZO5Mf+J1M5XcVN2J3g+yFYRw=	VtA6JWztUxlvRNvHlVK6JCamncua0L
ST012	R003	Pam Smith	BHMM3RZfhXeAzaIMAlDfGWfVDgtUsUvNOntgkmqfkk0=	pNzX5UULvKjKDdtt8s8vSHX1NO04AD
ST014	R002	Deidre Anne De Silva	nJotZzm/Y0RC1MnxokTik7cClWFOxSQBlYpyoHZ0puE=	hCelE6NsNLlxl5ZE1lhULPT7q3gdFD
ST016	R003	Nancy Meyer	SKeH+9rabiZxfnr+cYrH06S9pcVDHStEmYbzFy5O+OQ=	HgaFKaWVrvaUk7lKzGJCjgaQdicaSZ
ST023	R003	Rachel Green	dGwf5C/NTre1OAHmgUWv1Q9ayFbXnQu/8sOr4mSXdYw=	1SywZtSDBlKiWzgEPnBqWy7EbAh0W2
ST029	R001	Chloe Khoo	1RabwKZok9pQ+pgwkLzXJNuNti4qaE5Pf4M5W7DKxVg=	O2FMxUXPxpYaHQNdaYp7KTrEo7xd4r

Figure 5: Hashed passwords are stored in the database alongside the salt

The next security feature added to Care Foundation EMR system is audit logging. It allows the admin to maintain, monitor, and analyze the logs for any unusual activities. The audit log records all staff activities such as reading, updating, or deleting any records within the EMR system. Each log contains information such as the date and time, the action performed, and who performed the action. Figure 6 shows the interface design for the audit log.

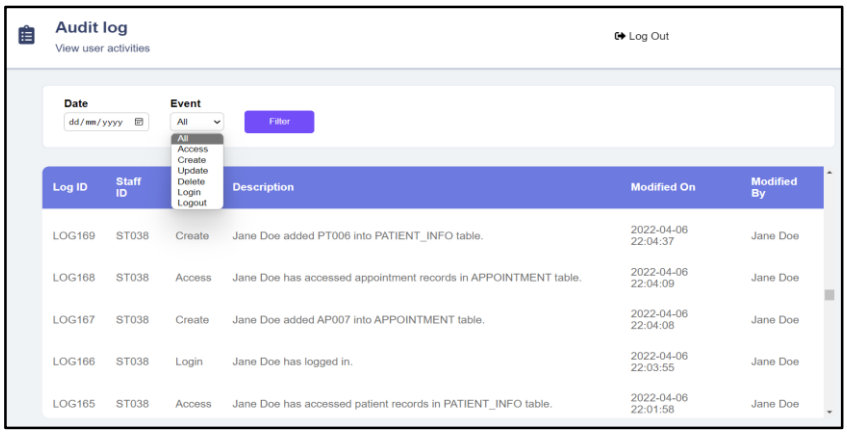


Figure 6: Interface for the audit log in the admin dashboard

Next, the encryption of all records is done using the Advanced Encryption Standard (AES) encryption algorithm. The following sections describe the implementation of AES encryption and decryption using the Java Cryptography Architecture (JCA) within the JDK. Figure 7 shows the function that generates the encryption key.

```
public String generateKey() {

    try {

        iv = new IvParameterSpec(initVector.getBytes("UTF-8"));
        key = KeyGenerator.getInstance("AES").generateKey();
        encodedKey = Base64.getEncoder().encodeToString(key.getEncoded());
        ecipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
        ecipher.init(Cipher.ENCRYPT_MODE, key, iv);

        return encodedKey;
    }
```

Figure 7: Code snippet for generating the encryption key

The mode of operation chosen is the Cipher Block Chaining (CBC), which uses an Initialization Vector (IV) to augment the encryption. For generating the secret key, the KeyGenerator class is used. The Cipher class is the one that handles the actual encryption and decryption, where an instance of it is created by passing a Cipher name as the parameter: “AES/CBC/PKCS5Padding”. Figures 8(a) and 8(b) show the function used to encrypt and decrypt sensitive data.

```
public String encrypt(String str) {
    try {

        byte[] bytes = ecipher.doFinal(str.getBytes());
        bytes = Base64.getEncoder().encode(bytes);
        String ciphertext = new String(bytes);

        return (ciphertext);
    }
```

```
public String decrypt(String str) {

    try {
        iv = new IvParameterSpec(initVector.getBytes("UTF-8"));
        dcipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
        dcipher.init(Cipher.DECRYPT_MODE, key, iv);

        byte[] base64decodedTokenArr = Base64.getDecoder().decode(str);
        byte[] decryptedText = dcipher.doFinal(base64decodedTokenArr);

        return new String(decryptedText);
```

Figure 8(a): Code snippet for data encryption, Figure 8(b): Code snippet for data decryption

The encrypt() function takes the string to be encrypted as the parameter, and returns the ciphertext of the string. The bytes are encoded with Base64 to ensure that it is intact without modification when being transferred. Similarly, the decryption of data works by passing the ciphertext to the doFinal() function. The encrypted string needs to be decoded first before performing the decryption process. Figure 9 shows the sensitive data in ciphertext after encryption and in plaintext after decryption.

Figure 9: The encrypted MEDICAL_RECORD table before and after decryption

5.2 Implementation of Functional Module

The implementation of the functional modules in Care Foundation EMR system is discussed in this section. Only a few important modules that are relevant to the EMR system are reviewed. The functional modules include the book appointment module, managing medical records module, and managing patient appointments module.

The patient portal is where the patient can book appointments for their next visit to the hospital. To do so, the patient needs to provide the appointment details and personal details. Figure 10 shows the code snippet for booking an appointment.

```
BookAppointment add_appt = new BookAppointment(appt_id, doc_id, enc.encrypt(patient_name),
        patient_email, enc.encrypt(patient_contact), appt_date, appt_time, status);
enc.addSecretKey(key_id, appt_id, secret_key);

int result = appt.bookAppt(add_appt);

if(result == 1){
    request.setAttribute("message", "success");
    request.setAttribute("appt_id", appt_id);
    request.getRequestDispatcher("make_appointment.jsp").forward(request,response);


}else{
    request.setAttribute("message", "error");
    request.getRequestDispatcher("make_appointment.jsp").forward(request,response);
}
```

Figure 10: Code snippet for booking an appointment

The patient also needs to complete the reCAPTCHA before they can proceed with the booking to prevent spam and abuse. If the booking is successful, the system allows the patient to save and print the letter of booking confirmation. Figures 11(a) and 11(b) show the interface during the appointment booking process.

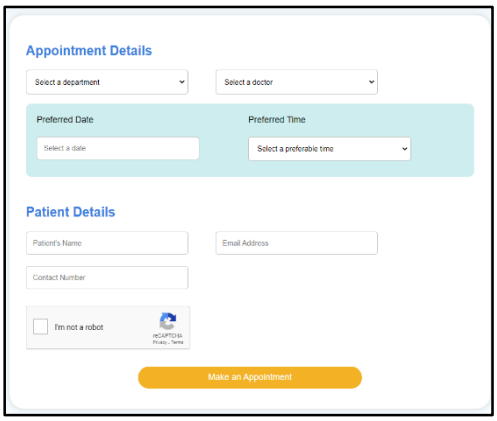


Figure 11(a): Interface for booking an appointment

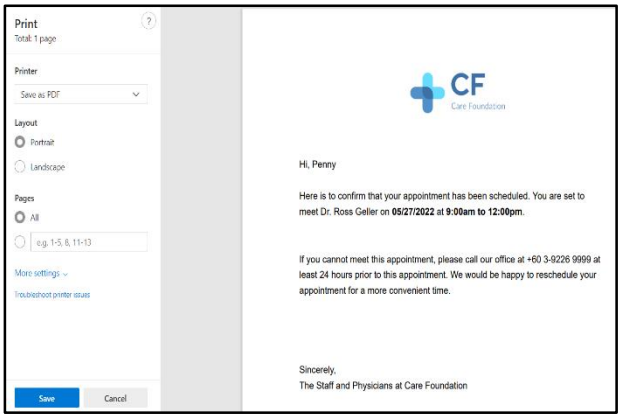


Figure 11(b): Interface for printing the appointment confirmation letter

The next functional module implemented in the EMR system is the manage medical records module. It allows doctors to easily manage patients' medical records, but only the records of patients that are assigned to them. Doctors can choose from a list of patients assigned to them to insert a medical record. They need to provide the abnormality of the patient, the treatment or medication prescribed to them, and optionally a remark if necessary. Once the process is successful, the action performed is logged and the system refreshes the page. Figure 12 shows the interface for adding a medical record for a patient.

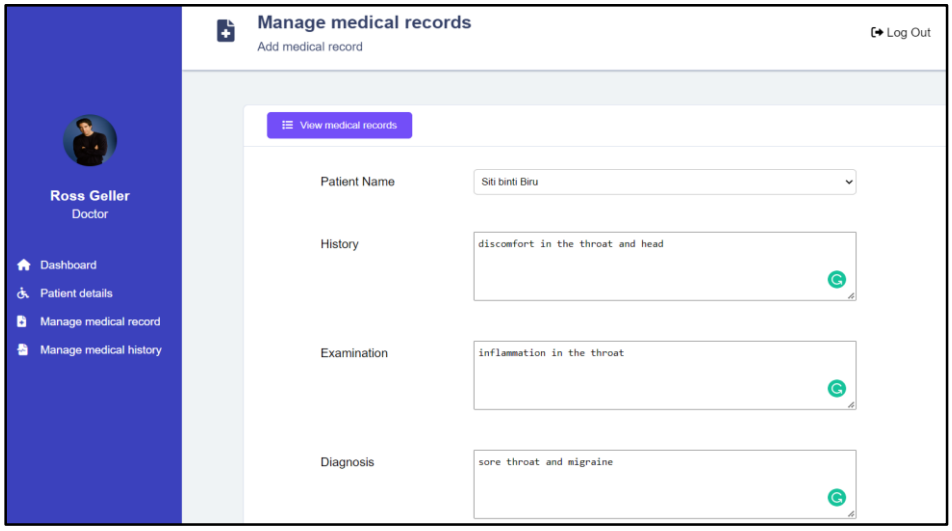


Figure 12: Interface for inserting medical records for patients.

Care Foundation EMR system also comes with a patient appointment management function that allows receptionists to add, update, or cancel patient appointments. This module lets the receptionists manually make appointments for patients who personally request them. To make an appointment, the system requires patient information such as name, email, and contact number, as well as the date and time of the appointment. Figure 13 shows the interface for viewing the list of appointments.

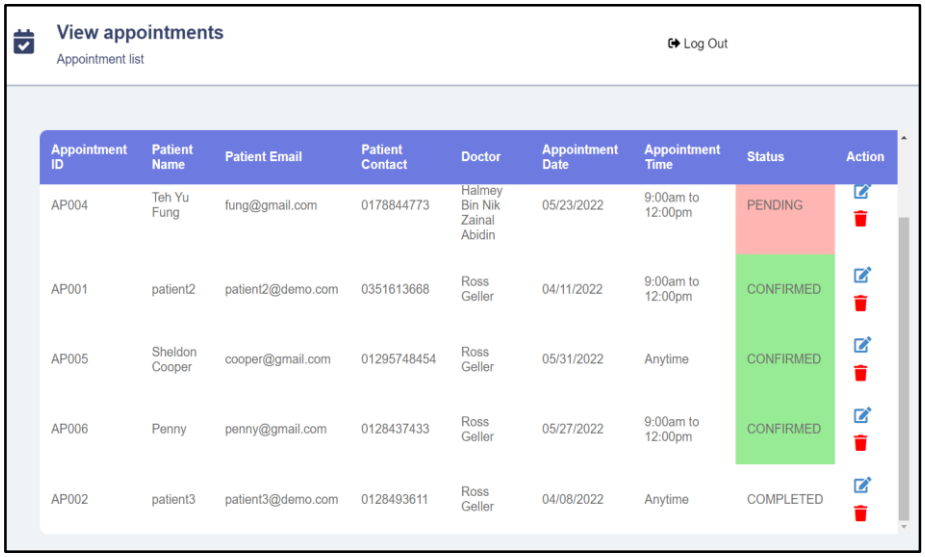


Figure 13: Interface for viewing the appointment list

5.3 System Testing

A test plan that consists of the functional and security requirements is designed to assess the functionality of the modules and the security features of Care Foundation EMR system. The goal of functional testing is to verify that the output of the developed system fulfils the functional requirements stated in the analysis phase. Security testing focuses on ensuring that the security mechanisms are integrated and working properly to safeguard the EMR system from security threats.

Table 6 tabulates the test results for the functional requirements of Care Foundation EMR system. In short, all the functional modules had passed the testing.

Table 6: Test results for functional requirements

Module	Test Case	Expected Output	Actual Output
Login	Login for the first time.	Redirects the user to the reset password page that forces them to change their password.	Pass
	Login when the account has been suspended.	Displays a warning message that says "Your account has been suspended."	Pass
User registration (admin, doctor, receptionist)	Using an existing email.	Displays a warning message that says "Email already exists!".	Pass
	Register with valid information.	Registers the user successfully and saves the information into the database.	Pass

Table 6: (cont)

Module	Test Case	Expected Output	Actual Output
Book appointment	Book an appointment with valid information.	Saves appointments into the database and lets the patient save the appointment confirmation letter.	Pass
Manage medical records	Add a medical record for a patient by selecting a patient's name.	Saves the medical record into the database and displays a success message.	Pass
	Update a medical record for a patient.	Only the abnormality, treatment, and remark can be updated, but not the patient's name. The updated record is successfully saved into the database.	Pass
Manage patient details	Using an existing email.	Displays a warning message that says "Email already exists!".	Pass
	Register a new patient with valid information.	Saves the patient details into the database and displays a success message.	Pass
	Update patient details.	Saves the updated patient details into the database and displays a success message.	Pass
Manage patient appointments	Book an appointment for a patient with valid information.	Saves the appointment into the database and displays a success message.	Pass
	Update patient appointment details.	Saves the updated patient details into the database and displays a success message.	Pass
	Cancel patient appointment.	Prompts the user before deleting. If the user clicks on "Yes", deletes the appointment from the database and displays a success message.	Pass

Table 7 tabulates the test results for the security requirements of Care Foundation EMR system. This test plan is designed to make sure that the security modules of the developed system are properly working to ensure security. In short, all the security modules had passed the testing.

Table 7: Test results for security requirements

Module	Test Case	Expected Output	Actual Output
Password policy	Set a password that does not fulfil the policy.	Displays a warning message that says "Password requirements not met!".	Pass
	New password and confirm password are not the same.	Displays a warning message that says "Passwords do not match!".	Pass
	New password is the same as the old one.	Displays a warning message that says "New password is the same as the old one!".	Pass
	Reset password that fulfils the requirement.	Updates the user password in the database and redirects to the dashboard.	Pass

Table 7: (cont)

Module	Test Case	Expected Output	Actual Output
Password hashing	Hash user passwords with salt during user registration.	Generates a salt and applies the hash algorithm on the password with the salt to generate a hashed password.	Pass
Authentication	Login using invalid credentials.	Displays an error message that says "Access denied!".	Pass
	Login using valid credentials.	Login is successful and the system redirects the user to the dashboard.	Pass
Account lockout policy	Login using invalid credentials for the third time.	Suspends the account and displays a warning message that says "Your account has been suspended."	Pass
	Login after having less than three unsuccessful attempts.	Resets the counter for the failed attempts. Login is successful and the system redirects the user to the dashboard.	Pass
Audit logging	Perform activities such as insert, update, delete, and view.	All activities are recorded in the database. The log provides an accurate description of the activity.	Pass
Automatic log-off	Leave the system idle for 5 minutes.	Automatically logs the user out of the account and redirects to the login page. The user has to log in again to access the system.	Pass
Data encryption	Encrypt sensitive data before saving it into the database.	Applies AES encryption to sensitive data before it is saved into the database. Sensitive data should be converted to ciphertext.	Pass
	Decrypt data to be viewed by the user.	Successfully decrypts the ciphertext so that it can be viewed by an authorized user.	Pass

5.4 User Acceptance Testing

The user acceptance test was carried out by inviting medical professionals and receptionists in the healthcare industry to test out the system. They were given a list of tasks to be performed while interacting with the EMR system, and once they are done, the users were told to fill in a Google Form where they provide their thoughts and feedback on the system. There were 7 doctors and 5 receptionists involved in the testing. The results from the forms are analyzed and tabulated in Table 8, Table 9, and Table 10.

Table 8: Result for system function testing (doctor)

Functional Module	Scale					Total
	1	2	3	4	5	
I am able to log into the system using the correct email and password.	0	0	0	2	5	7
When logging in for the first time, I am able to reset my password.	0	0	0	1	6	7
I am able to view the personal details of the patients that are assigned to me.	0	0	0	3	4	7
I am able to add a medical record for the patients assigned to me.	0	0	0	1	6	7
I am able to view the patients' medical records created by me.	0	0	0	4	3	7
I am able to update the patients' medical records.	0	0	0	2	5	7
I am able to add a medical history record for the patients assigned to me.	0	0	0	3	4	7
I am able to view the patients' medical history created by me.	0	0	0	0	7	7
I am able to update the patients' medical history records.	0	0	0	3	4	7
I am able to log out of the system and be redirected back to the staff login page.	0	0	0	1	6	7

Table 9: Result for system function testing (receptionist)

Functional Module	Scale					Total
	1	2	3	4	5	
I am able to log into the system using the correct email and password.	0	0	0	1	4	5
When logging in for the first time, I am able to reset my password.	0	0	0	2	3	5
I am able to register a patient by filling in the details.	0	0	0	1	4	5
I am able to view the patients' personal details created by me.	0	0	0	2	3	5
I am able to update the patients' personal details.	0	0	0	0	5	5
I am able to book an appointment for a patient.	0	0	0	2	3	5
I am able to view the list of patient appointments made by me.	0	0	0	2	3	5
I am able to update the patient appointment details.	0	0	0	2	3	5
I am able to cancel a patient's appointment.	0	0	0	2	3	5
I am able to log out of the system and be redirected back to the staff login page.	0	0	0	2	3	5

Table 10: Result for user experience

Items	Scale					Total
	1	2	3	4	5	
I find the system easy to use.	0	0	1	3	8	12
The system requires the fewest steps possible to accomplish what I want to do with it.	0	0	1	4	7	12
The usage of terms throughout the system is consistent.	0	0	1	2	9	12
The positioning of messages across the screen is consistent.	0	0	1	4	7	12
The prompts displayed for inputs are clear.	0	0	0	1	11	12
I find it easy to navigate the system.	0	0	0	6	6	12
I am satisfied with the interface design of the system.	0	0	1	4	7	12
Overall, I am satisfied with this system.	0	0	1	5	6	12

The overall results of the user acceptance test showed that Care Foundation EMR system is generally well-accepted by the end-users.

6. Conclusion

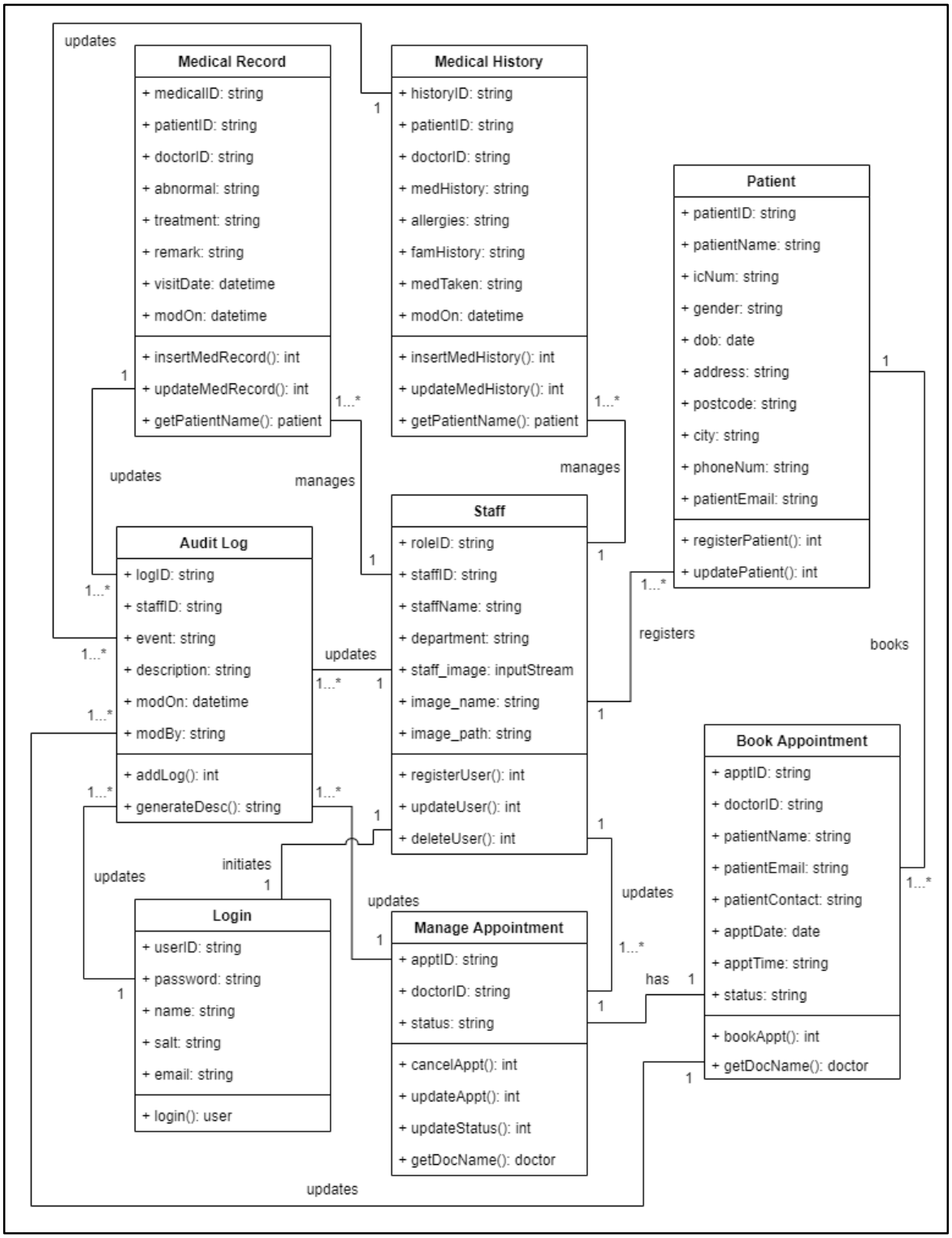
To summarize, the proposed system can transform the functional and security requirements defined in the analysis phase based on the designs in the design phase. In terms of functionality, Care Foundation

EMR system allows healthcare staff to quickly access patient information and streamline several routine tasks, which in turn increases productivity and efficiency. Apart from this, the system is also able to integrate the appropriate security techniques that safeguard sensitive patient information stored in the database using cryptography and control the access of hospital staff by restricting the EMR system to users in a certain role to prevent unauthorized access. The system also allows administrators to monitor, track, and audit the activities of every user within the system. However, the system is not without flaws. For example, patients can only browse the patient portal in desktop mode as there is no support yet for mobile devices. Additionally, more advanced features such as telemedicine, medical billing, internal messaging, and e-prescribing are not yet available in the proposed system. Thus, future work may consider expanding the scope of development, such as making the patient portal more accessible via mobile devices. Furthermore, it would be interesting to incorporate assistive technology into the website, which would be helpful to people with disabilities or impairments. An accessible website could encourage people with special needs to participate more actively.

Acknowledgment

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support and encouragement throughout the process of conducting this study.

Appendix A


Medical Record
+ medicalID: string
+ patientID: string
+ doctorID: string
+ abnormal: string
+ treatment: string
+ remark: string
+ visitDate: datetime
+ modOn: datetime
+ insertMedRecord(): int
+ updateMedRecord(): int
+ getPatientName(): patient
Medical History
+ historyID: string
+ patientID: string
+ doctorID: string
+ medHistory: string
+ allergies: string
+ famHistory: string
+ medTaken: string
+ modOn: datetime
+ insertMedHistory(): int
+ updateMedHistory(): int
+ getPatientName(): patient
Patient
+ patientID: string
+ patientName: string
+ icNum: string
+ gender: string
+ dob: date
+ address: string
+ postcode: string
+ city: string
+ phoneNum: string
+ patientEmail: string
+ registerPatient(): int
+ updatePatient(): int
Staff
+ roleID: string
+ staffID: string
+ staffName: string
+ department: string
+ staff_image: inputStream
+ image_name: string
+ image_path: string
+ registerUser(): int
+ updateUser(): int
+ deleteUser(): int
Audit Log
+ logID: string
+ staffID: string
+ event: string
+ description: string
+ modOn: datetime
+ modBy: string
+ addLog(): int
+ generateDesc(): string
Book Appointment
+ apptID: string
+ doctorID: string
+ patientName: string
+ patientEmail: string
+ patientContact: string
+ apptDate: date
+ apptTime: string
+ status: string
+ bookAppt(): int
+ getDocName(): doctor
Login
+ userID: string
+ password: string
+ name: string
+ salt: string
+ email: string
+ login(): user
Manage Appointment
+ apptID: string
+ doctorID: string
+ status: string
+ cancelAppt(): int
+ updateAppt(): int
+ updateStatus(): int
+ getDocName(): doctor
updates
manages
registers
books
initiates
has
1
1...*

References

[1] M. A. Rothstein, "Health Privacy in the Electronic Age," J. Leg. Med., vol. 28, no. 4, pp. 487–501, Oct. 2007, doi: 10.1080/01947640701732148.

[2] T. Greenhalgh, S. Hinder, K. Stramer, T. Bratan, and J. Russell, "Adoption, non-adoption, and abandonment of a personal electronic health record: case study of HealthSpace," BMJ, vol. 341, 2010, doi: 10.1136/bmj.c5814.

[3] A. Boonstra and M. Broekhuis, "Barriers to the acceptance of electronic medical records by physicians from systematic review to taxonomy and interventions," BMC Health Serv. Res., vol. 10, no. 1, pp. 1–17, Aug. 2010, doi: 10.1186/1472-6963-10-231/TABLES/3.

[4] W. Hurst, A. Boddy, M. Merabti, and N. Shone, "Patient Privacy Violation Detection in Healthcare Critical Infrastructures: An Investigation Using Density-Based Benchmarking," Future Internet, vol. 12, no. 6, 2020, doi: 10.3390/fi12060100.

[5] A. S. Al-Adwan and H. Berger, "Exploring physicians' behavioural intention toward the adoption of electronic health records: An empirical study from Jordan," International Journal of Healthcare Technology and Management, vol. 15, no. 2, pp. 89–111, 2015, doi: 10.1504/IJHTM.2015.074538.

[6] "June 2020 Healthcare Data Breach Report.", Accessed: Dec. 23, 2021. [Online]. Available: https://www.hipaajournal.com/june-2020-healthcare-data-breach-report/

[7] K. Caine and R. Hanania, "Patients want granular privacy control over health information in electronic medical records," J. Am. Med. Informatics Assoc., vol. 20, no. 1, pp. 7–15, Jan. 2013, doi: 10.1136/AMIAJNL-2012-001023/3/AMIAJNL2012001023F01.JPEG.

[8] W. C. Chao, H. Hu, C. O. L. Ung, and Y. Cai, "Benefits and Challenges of Electronic Health Record System on Stakeholders: A Qualitative Study of Outpatient Physicians," J. Med. Syst. 2013 374, vol. 37, no. 4, pp. 1–6, Jul. 2013, doi: 10.1007/S10916-013-9960-5.

[9] C. C. Sines and G. R. Griffin, "Potential Effects of the Electronic Health Record on the Small Physician Practice: A Delphi Study," Perspect. Heal. Inf. Manag., vol. 14, no. Spring, Mar. 2017, Accessed: Dec. 23, 2021. [Online]. Available: /pmc/articles/PMC5430134/.

[10] N. Aissaoui, M. Aissaoui, Y. J.-I. J. of Computer, and undefined 2013, "For a cloud computing based open source E-health solution for emerging countries," Citeseer, vol. 84, no. 11.

[11] B. Jones, X. Yuan, E. Nuakoh, and K. Ibrahim, "Survey of Open Source Health Information Systems," Health Inform, vol. 3, no. 1, pp. 23–31, Feb. 2014, doi: 10.5121/HIIJ.2014.3102.

[12] E. Helms and L. Williams, "Evaluating access control of open source electronic health record systems," Proceedings of the 3rd Workshop on Software Engineering in Health Care, no. 11, pp. 63–70, 2011, doi: 10.1145/1987993.1988006.

[13] N. Mohammed, A. Munassar, P. Student, and A. Govardhan, "Comparison between Traditional Approach and Object-Oriented Approach in Software Engineering Development," International Journal of Advanced Computer Science and Applications, vol. 2, no. 6, 2011, Accessed: Nov. 23, 2021. [Online]. Available: www.ijacsa.thesai.org.

[14] J. Sun and Y. Fang, "Cross-domain data sharing in distributed electronic health record systems," IEEE Transactions on Parallel and Distributed Systems, vol. 21, no. 6, pp. 754–764, 2010, doi: 10.1109/TPDS.2009.124.

[15] A. A. Zaidan, B. B. Zaidan, A. Al-Haiqi, M. L. M. Kiah, M. Hussain, and M. Abdulnabi, "Evaluation and selection of open-source EMR software packages based on integrated AHP and TOPSIS," J. Biomed. Inform., vol. 53, pp. 390–404, Feb. 2015, doi: 10.1016/J.JBI.2014.11.012.

[16] D. Uhlman "Introduction to ClearHealth – OSnews.", 2015 Accessed: Nov. 6, 2021. [Online]. Available: https://www.osnews.com/story/10740/

[17] J. Lake X. Yuan and J. Zhan "Towards authentication vulnerabilities in openemr" The 2013 Symposium on Computing at Minority Institutions April 2013.

[18] M. Yi, "Major Issues in Adoption of Electronic Health Records Subject Categories and Descriptors J.3 [LIFE AND MEDICAL SCIENCES]; Medical information systems," Journal of Digital Information Management, vol. 16, 2018, doi: 10.6025/jdim/2018/16/4/180-191.

[19] J. Cleland-Huang, A. Czauderna, M. Gibiec, and J. Emenecker, "A Machine Learning Approach for Tracing Regulatory Codes to Product Specific Requirements," in Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 1, 2010, pp. 155–164, doi: 10.1145/1806799.1806825.

[20] Z. Whittaker, "Thousands of US lab results and medical records spilled online after a security lapse", TechCrunch, 2020. Accessed: Dec. 22, 2021. [Online]. Available: https://techcrunch.com/2020/12/01/ntreatment-lab-results-medical-records-exposed/

[21] M. K. McGee, "Electronic Health Records: Spotlighting Risks", Healthcare Info Security, 2020. Accessed: Dec. 22, 2021. [Online]. Available: https://www.healthcareinfosecurity.com/electronic-health-records-spotlighting-risks-a-15525

[22] S. Alder, "Vanderbilt University Medical Center Employees Inappropriately Accessed 3,000 Patients' PHI", HIPAA Journal, 2017. Accessed: Dec. 22, 2021. [Online]. Available: https://www.hipaajournal.com/vanderbilt-university-medical-center-employees-found-to-have-inappropriately-accessed-300-patients-phi-8709/

[23] T. Wilson, "Insider May Have Breached More Than 10,000 Patient Records At Johns Hopkins", Dark Reading, 2009. Accessed: Jul. 21, 2022. [Online]. Available: https://www.darkreading.com/risk/insider-may-have-breached-more-than-10-000-patient-records-at-johns-hopkins

[24] L. Whitney, "Healthcare organizations targeted with password spraying attacks", Tech Republic, 2020. Accessed: Jul. 21, 2022. [Online]. Available: https://www.techrepublic.com/article/healthcare-organizations-targeted-with-password-spraying-attacks/

[25] D. Brown, "Executives' weak passwords lead to breaches in healthcare and other industries", McKnights, 2022. Accessed: Jul. 21, 2022. [Online]. Available: https://www.mcknights.com/news/executives-weak-passwords-lead-to-breaches-in-healthcare/