



**UNIVERSITÀ
DEGLI STUDI
DI PADOVA**



DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

**CORSO DI LAUREA MAGISTRALE IN
ICT FOR INTERNET AND MULTIMEDIA - INGEGNERIA PER LE
COMUNICAZIONI MULTIMEDIALI E INTERNET**

**“ZERO-KNOWLEDGE ARCHITECTURE FOR SECURE HEALTHCARE DATA
SHARING”**

Relatore: Prof. ALEXANDRU GRADINARU

Laureando/a: FARNOOSH ASLANZADEH

Correlatore: Prof./Dott GIULIA CISOTTO

ANNO ACCADEMICO 2023-2024

Data di laurea 11/07/2024

Table of Contents

LIST OF FIGURES 4

ABSTRACT..... 5

1. INTRODUCTION..... 7

1.1. THE STRUCTURE OF THE PAPER 8

2. STATE OF THE ART 9

2.1. INTRODUCTION..... 10

2.2. NETWORK ARCHITECTURE AND FILE STORAGE 10

2.3. DECENTRALIZED AND DISTRIBUTED ARCHITECTURES..... 10

2.4. HYBRID STORAGE APPROACHES..... 11

2.5. DISTRIBUTED FILE SYSTEMS AND INTERPLANETARY FILE SYSTEM (IPFS) 11

2.6. PASSWORD MANAGEMENT AND SECURE DATA SHARING SYSTEMS 13

2. DOCUMENT SHARING AND ACCESS CONTROL 13

2.1. ELECTRONIC HEALTH RECORD (EHR) SHARING FRAMEWORKS 13

2.2. FINE-GRAINED ACCESS CONTROL MECHANISMS 13

2.3. INTEROPERABILITY AND STANDARDIZATION 14

3. ACTION LOGGING AND PERFORMANCE 14

3.1. BLOCKCHAIN-BASED ACTION LOGGING 14

3.2. PERFORMANCE EVALUATION AND OPTIMIZATION 14

3.3. HYBRID APPROACHES AND INTEGRATION WITH EXISTING SYSTEMS 15

3.4. COMPARATIVE ANALYSIS 15

3.5. CONCLUSION 16

3.5. HIGHLIGHT 17

4. PROPOSED SOLUTION..... 18

4.1. INTRODUCTION 18

4.2. SOFTWARE SPECIFICATION... 19

4.2.1. PROBLEM ANALYSIS 19

4.3. SCENARIOS 20

4.4. USER TYPES (ACTORS)..... 24

4.5. USE CASES PRESENTED IN DETAIL..... 26

4.5.1 INTRODUCTION.....	26
4.6. USE CASE DIAGRAM WITH ALL FUNCTIONALITIES	28
4.7. INTERFACE MOCKUPS	30
5. ARCHITECTURAL DESIGN.....	32
5.1. INRTRODUCTION.....	32
5.2. DESCRIBE THE HIGH-LEVEL COMPONENTS	32
5.3. DEPLOYMENT + COMPONENTS DIAGRAM	32
5.4. DATABASE STRUCTURE (ERD DIAGRAM + DETAILS) / FILE STORAGE.....	35
5.5. SECURITY	37
6. IMPLEMENTATION DETAILS.....	40
6.1. INTRODUCTION.....	40
6.2. LOGIC APP	41
6.3. STORE ENCRYPTED LOGIC APP	44
6.4. USER-FRIENDLY WEB PORTAL FOR PATIENT.....	55
6.4. TECHNICAL IMPLEMENTATION	55
6.4. USER-FRIENDLY WEB PORTAL FOR PHARMACY	56
6.4.1. TECHNICAL IMPLEMENTATION	57
7. RESULTS	60
7. 1. INTRODUCTION.....	60
7. 2. TEST SCENARIOS.....	60
7.2.1. UNIT TEST	60
7.2.2. INTEGRATION TEST.....	62
7.2.3 EVALUATION OF THE PERFORMANCE	63
7.3.USER STORY STATEMENT	64
7.4. SECURITY KEY POINT.....	64
7.5. SOME FEEDBACK.....	65
7.6.KEY SECURITY HIGHLIGHT	66
7.7. FRONT-END APPLICATION.....	66
8. CONCLUSIONS	70
9. REFRENCES	72

LIST OF FIGURES

Figure 1 21

Figure 2 22

Figure 3 23

Figure 4 28

Figure 5 31

Figure 6 34

Figure 7 35

Figure 8 37

Figure 9 37

Figure 10 41

Figure 11 48

Figure 12 50

Figure 13 51

Figure 14 46

Figure 15 54

Figure 16 54

Figure 17 67

Figure 18 67

Figure 19 68

Figure 20 69

ABSTRACT

This thesis addresses the urgent need for a safe effective and efficient exchange of patient data between healthcare providers by investigating novel approaches to secure document sharing in the field. We offer a novel method for managing secure document sharing between physicians and patients that makes use of integrated cloud services. This approach is contrasted with blockchain-based solutions to assess the advantages and disadvantages of each. The study examines decentralized and centralized blockchain strategies while taking patient control data integrity scalability and performance into account. Our study attempts to shed light on the best architectural solutions for the sharing of medical records while taking the intricate needs of data security accessibility and legal compliance into account. A working prototype and several usage scenarios are also presented. The results may help healthcare organizations choose the right document sharing systems and add to the continuing conversation about health IT optimization.

ABSTRACT IN ITALIAN:

Questa tesi affronta l'urgente necessità di uno scambio sicuro, efficace ed efficiente dei dati dei pazienti tra operatori sanitari studiando nuovi approcci per proteggere la condivisione dei documenti sul campo. Offriamo un nuovo metodo per gestire la condivisione sicura dei documenti tra medici e pazienti che si avvale di servizi cloud integrati. Questo approccio è in contrasto con le soluzioni basate su blockchain per valutare i vantaggi e gli svantaggi di ciascuna. Lo studio esamina le strategie blockchain decentralizzate e centralizzate tenendo conto della scalabilità e delle prestazioni dell'integrità dei dati di controllo dei pazienti. Il nostro studio tenta di far luce sulle migliori soluzioni architettoniche per la condivisione delle cartelle cliniche tenendo conto delle complesse esigenze di accessibilità alla sicurezza dei dati e di conformità legale. Vengono inoltre presentati un prototipo funzionante e diversi scenari di utilizzo. I risultati potrebbero aiutare le organizzazioni sanitarie a scegliere i giusti sistemi di condivisione dei documenti e contribuire al dibattito continuo sull'ottimizzazione dell'IT sanitario.

1. INTRODUCTION

The secure sharing of health documents presents complex technical challenges that require innovative architectural solutions. This analysis explores two contrasting approaches: a centralized cloud-based system and a decentralized blockchain-based framework, each with distinct advantages and limitations for managing sensitive health data [3]. A centralized cloud solution uses a range of integrated services to create a robust and scalable document sharing infrastructure [2,4].

Essentially, a centralized cloud provides globally distributed storage for encrypted documents, ensuring high availability and low latency across geographic regions. This is critical in healthcare environments where quick access to patient data can be critical. The system uses centralized encryption key management that improves security with features such as automatic key rotation and access logging [5,8]. In contrast, a blockchain-based solution offers better data integrity, transparency and patient management due to its decentralized architecture [3,6,9]. It has challenges in terms of performance, scalability and integration with existing systems [3,7]. Both approaches offer unique advantages and limitations in meeting the complex requirements of health record sharing. The optimal choice depends on specific organizational needs, technical capabilities and regulatory considerations, and possible hybrid solutions combine elements of both approaches [4, 7].

Meanwhile, in the prevailing pharmaceutical prescription workflow, patients often visit a doctor, who gives them a prescription that they take to a pharmacy. Frequently this normally necessitates having to physically move documents through paper or use inappropriate digital means thus posing privacy, trustworthiness and raising concern about data privacy. Many major pharmacy chains continue to use traditional systems for handling prescriptions as shown by a study, which are prone to mistakes and delays besides exposing patient information to potential hacking threats online. Traditional system has medication error and lack of productivity and increase the cost.

Besides, zero-knowledge architectures that are emerging offer an exciting option for secure healthcare data sharing. This type of system allows for privacy-preserving computations and enables parties to prove things about data without revealing the data itself. By using advanced cryptographic techniques, zero-knowledge solutions could potentially tackle some of the

limitations of both centralized and decentralized systems thereby making it possible for secure, scalable and accessible healthcare document control to take on a new paradigm. Moreover, further research and development in this field may result to innovative hybrid frameworks that integrate cloud computing, blockchain and zero-knowledge approaches thus enabling the next wave of IT infrastructure for healthcare.

In this thesis, we propose a new method for secure sharing of documents between patients and healthcare professionals using Azure cloud services. Our approach uses integrated cloud technologies to solve some of the complex problems of secure healthcare management such as sharing the prescription securely and is managed by the patient to share with authorized pharmacies. The results of this new method are compared with blockchain-based solutions, providing a comprehensive analysis of their strengths and weaknesses in the context of sharing health records.

1.1 The structure of the paper:

This paper is pointing out the importance of securely sharing health records and introduces two different approaches; one is centralized cloud-based system, and another is decentralized blockchain-based framework. The next part of this document explains in detail both these methods, stating their advantages and disadvantages which have been explained in the state of the art . These papers' core involves the suggested strategy that employs Azure's cloud services to enable physicians and patients to share documents in a secure manner that it is going discussed in the action logs and the performance part. In the implementation part, describes how integrated Azure services are used in explaining complex healthcare security issues by details. Finally, the paper wraps up by highlighting key features as well as benefits of proposed approach based on Azure technology in the result, with an insight into future improvement designs which include characteristics of centralized portal architectures in the conclusion section.

2.STATE OF THE ART:

2.1 Introduction:

Security, privacy interoperability and data integrity are among the major issues in healthcare data management that blockchain technology has shown promise in resolving. The key areas of network architecture, file storage document sharing, access control, action logging and performance are the focus of this review which looks at the state-of-the-art in adding blockchain to healthcare systems.

The initial segment delves into diverse methods for refining network architecture and data storage in blockchain-driven healthcare systems. To efficiently manage large medical files, it covers decentralized and distributed architectures hybrid storage approaches that combine on-chain and off-chain solutions and the integration of distributed file systems like IPFS (Interplanetary File System) [8]. The second section explores fine-grained access control and secure document sharing frameworks and mechanisms. This study looks at advanced access control mechanisms blockchain-based Electronic Health Record (EHR) sharing systems and standardization initiatives aimed at achieving interoperability. The use of blockchain for action logging in the healthcare industry is covered in the final section which emphasizes how it can be used to create auditable and unchangeable records of important events. In order to optimize blockchain-based systems for healthcare applications it also covers performance considerations and the need for additional research.

By highlighting significant obstacles and future development opportunities in this quickly developing field this review seeks to give academics and industry professionals a thorough understanding of the most recent developments and limitations in blockchain technology for the healthcare industry.

2.2 NETWORK ARCHITECTURE AND FILE STORAGE

Blockchain technology has the potential to significantly change how medical data is shared, stored and preserved. Blockchain-based healthcare systems must however be successful which means selecting the appropriate file storage technologies and designing a scalable effective network landscape. Numerous scholars have explored diverse approaches for enhancing network architecture and data storage to enhance security scalability and responsiveness.

2.3 Decentralized and Distributed Architectures

Blockchain technology has the potential to significantly alter data sharing and medical record archiving. But the selection of appropriate file storage technology and the development of a practical scalable network architecture are ultimately responsible for the success of blockchain-based healthcare systems. Several researchers have looked into various methods for streamlining network architecture and file storage in order to improve security scalability and responsiveness. To solve these problems Abunadi and Kumar [10] developed the Blockchain Security Framework for Electronic Health Records (BSF-EHR) which uses a decentralized network architecture. In their model the authors stress the significance of implementing access control measures to guarantee that patient data is only accessible by authorized individuals. Based on its own blockchain every node within the BSF-EHR system has an access control system. Only the EHRs for which they have been granted access can be viewed by patients' insurance agents within their own blockchain replica. Additionally, the BSF-EHR encrypts EHRs using the patient's public key prior to adding them to blocks. Only authorized medical professionals and insurance representatives have access to the private keys required for decryption. Additionally bilinear maps are produced that combine patient IDs with the encrypted medical records after the encrypted EHRs have been hashed. Data integrity checking and extra security layers are provided. BSF-EHR uses a decentralized blockchain architecture with different node types of granular access control and cryptographic techniques to support safe and efficient EHR sharing between patients, doctors and payers while protecting patient privacy. The findings demonstrate that BSF-EHR can successfully secure EHR data in contrast to centralized approaches.

BSF-EHR model, though has the security and privacy aspects, would require a practical consideration of its limitations and operational realities in large scale healthcare settings for a full assessment to be made on whether the concept is viable or not.

2.4 Hybrid Storage Approaches:

Blockchain technology offers a secure platform for data storage, but scalability problems and privacy concerns prevent it from being used with large files. Researchers have suggested hybrid storage solutions that integrate off-chain and on-chain storage technologies in order to overcome these issues. In relation to data storage Cernian et al. additionally to Lee et al. Used a hybrid strategy in their blockchain-based personal health record (PHR) systems [5]. Sensitive PHR data is encrypted and kept in a safe off-chain database with Patient Data Chain [4]. In order to guarantee the data integrity and immutability the blockchain keeps a hash of it. Maintain encrypted PHR data in a safe database and control access and sharing permissions by leveraging the blockchain [5]. Hailemichael et al. combined off-chain and on-chain storage. demonstrate the feasibility of conducting statistical analysis on distributed EHR data while maintaining privacy. A virtual dataset is created by running searches on disparate EHR systems and storing the compiled results for safe analysis on the blockchain.

2.5 Distributed File Systems and InterPlanetary File System (IPFS)

Using distributed file systems such as IPFS (Interplanetary File System) [8] to alleviate storage constraints in blockchain networks has been studied by a few researchers. IPFS is a peer-to-peer distributed file system that stores and retrieves large files quickly by using content-addressed hashing. Zolfaghari along with others. To handle large medical files such as genetic and imaging data [15] recommend combining IPFS with blockchain-based healthcare systems. They store the files themselves on IPFS and the hash values on the blockchain respectively according to their methodology. It makes data integrity guaranteed and secure access control possible. Medical record sharing is essential to improve supporting patient and speed up medical procedures. Data integrity auditable document transfers and access control management are two areas where blockchain technology has a lot of abilities .

2.6 PASSWORD MANAGEMENT AND SECURE DATA SHARING SYSTEMS

1Password [1] protects user data by using an advanced end-to-end encryption model. In essence it derives encryption and authentication keys using a process called two-secret key derivation (2SKD) which combines a high-entropy Secret Key with an account password that the user has memorized. Even if server data is compromised this method stops offline password guessing attempts. The system makes use of contemporary cryptographic primitives such as the Secure Remote Password (SRP) protocol for zero-knowledge authentication and AES-256-GCM for authenticated encryption. Key exchanges between users and groups can be done securely thanks to public key cryptography. True end-to-end encryption, the servers ignorance of user secrets and server-stored uncrackable verifiers are important security features. There are several tiers of transport security in place such as TLS and extra application-layer encryption.

In addition to server and client policy enforcement the design places a strong emphasis on the cryptographic enforcement of access controls whenever it is feasible. To strike a balance between security and usability meticulous key management and secure recovery procedures are used. All things considered, the 1Passwords method constitutes a cutting-edge solution for safely exchanging and storing private information in a cloud environment with multiple users while retaining a high level of usability.

Furthermore, 1Password security design is highly adaptable and scalable, using strong cryptography such as two-step key derivation and secure remote password authentication to keep user data safe even when the server is breached. The system is built with high security features in mind that are capable of accommodating growth; this means that no infiltrators can access data.

This model offers a zero-knowledge architecture example that can be replicated for healthcare data since the data encryption is crucial for 1Password and is a good model for making the patients as a center of the system to record their data and also user can easily manage the authorized user to access the data.

2. DOCUMENT SHARING AND ACCESS CONTROL:

2.1 Electronic Health Record (EHR) Sharing Frameworks

Blockchain technology has been proposed by several frameworks and architectures for the distribution of electronic health records (EHRs). Dubovitska et al. [11]. introduce ACTION-EHR a patient-centered blockchain system made to make it easier for people to organize and share electronic health records when receiving cancer therapy. While maintaining metadata on-chain encrypted EHR data is stored off-chain in a cloud-based storage system with ACTION-EHR utilizing a hybrid data management strategy. Furthermore, the architecture of OmniPHR which permits the secure exchange of patient health records (PHRs) between various healthcare providers is explained by Roehrs et al. [14]. OmniPHR facilitates data sharing and interoperability between various blockchain networks by utilizing a distributed peer-to-peer network in conjunction with a middleware-based approach.

2.2 Fine-grained Access Control Mechanisms

Researchers have investigated several blockchain-based access control techniques in an effort to guarantee the integrity and confidentiality of shared medical records. Hussein. offer a framework for combining attribute-based access control with smart contracts to enable dynamic and granular access control for EHRs [13]. Their methodology states that smart contracts executed on the blockchain are used to enforce access policies created with attributes pertaining to user's objects and environmental conditions.

2.3 Interoperability and Standardization

Interoperability between various blockchain-based document sharing systems and the current medical IT infrastructure must be achieved for blockchain technology to be widely used in the medical field. Zhang addresses this issue by offering a framework that combines the Fast Healthcare Interoperability Resources (FHIR) standard with blockchain technology [3,6]. Their approach facilitates seamless integration with current EHR systems and makes document sharing across organizations simple. Standard data formats and protocols like FHIR and HL7 must be adopted for blockchain-based document sharing systems to be compatible and interoperable [12][13]. The effective sharing of medical data will be facilitated by encouraging blockchain networks and healthcare organizations to abide by these rules.

3. Action Logging and Performance

Event recording in healthcare applications is a perfect fit for blockchain technology since it provides an auditable and immutable record of all actions made on the system. By recording critical events such as patient consent data access and medical record modifications on the blockchain healthcare organizations can ensure their integrity and prevent them from being reversed.

3.1 Blockchain-based Action Logging

Dubovitskaya et al [2] show off the benefits and suitability of blockchain technology for activity logging in medical settings. The article additionally suggested managing patient access and consent in the context of diabetes care using blockchain technology. Using blockchain technology their system logs patient consent and data access events producing an auditable and unchangeable record of each event. Moreover, a blockchain-based framework for the safe and dependable exchange of radiation oncology-related EHR data is offered. Their system guarantees accountability and transparency by utilizing smart contracts to log data exchange events and access control events on the blockchain.

3.2 Performance Evaluation and Optimization

Although blockchain technology has advantages over traditional database or file-based storage systems its usefulness in the healthcare industry is still unknown. Proposing a blockchain-based model that uses proxy re-encryption for data sharing Thwin and Vasupongayya [16] offer granular access control over medical records kept in cloud storage. Researchers should investigate ways to improve storage efficiency in order to optimize blockchain systems for the healthcare industry. Massive logs may need to be stored off-chain on IPFS-like systems to preserve log integrity and verifiability through an on-chain hash system. Furthermore, when handling substantial amounts of log data the blockchain network performance can be enhanced by utilizing sharding and other scalability strategies. These methods might assist in resolving scalability issues while maintaining blockchains advantages in terms of security and privacy for healthcare applications.

3.3 Hybrid Approaches and Integration with Existing Systems

Hybrid approaches which combine blockchain technology with other distributed technologies such as distributed databases are being studied by researchers to leverage the advantages of blockchain-based storage while addressing performance issues. Take Cernian et al. for example. In their analysis of the PatientDataChain system [4] (2020) showed how production efficiency can be increased while maintaining the three main advantages of blockchain technology: transparency immutability and decentralization. Integration with the current healthcare IT infrastructure is necessary for the successful deployment of blockchain-based registration systems. In order to enable data sharing between blockchain networks and traditional systems such as data analysis platforms and EHR clinical decision support systems, standard interfaces and APIs must be established.

3.4 Comparative Analysis:

A comparison of blockchain-based healthcare system solutions reveals a variety of creative strategies each with advantages and disadvantages of their own. Abunadi and Kumar [10] created the BSF-EHR framework which has a decentralized architecture granular access control and cutting-edge encryption methods. When working with massive volumes of EHR data it might encounter scalability issues and its encryption and bilinear mapping procedures might demand a lot of processing power.

Alternatively, the Patient Data Chain system [4] uses a hybrid storage strategy leveraging the blockchain to preserve data integrity while off-chaining sensitive data storage. While there is a chance that this approach will increase scalability it also poses new difficulties in maintaining consistency between off-chain and on-chain data and can cause performance snags when accessing off-chain data often.

In an attempt to manage massive medical files more effectively some researchers [15] have looked into combining blockchain technology with distributed file systems such as IPFS. Large file storage and retrieval are facilitated by this method, but it also adds external network dependencies and could make the system architecture more complex overall.

Blockchain technology is being used in specialized healthcare domains as evidenced by the ACTION-HER [11] system which was created especially for cancer treatment scenarios. It makes

use of cloud-based storage in conjunction with a hybrid data management strategy. However, this dependence on cloud infrastructure may raise more security and privacy issues.

Using a middleware-based strategy OmniPHR [14] adopts a different approach to facilitate interoperability across various blockchain networks. Although this increases flexibility it also complicates the system architecture and could result in performance overhead.

Some solutions like the smart contract-based access control system put forth by Hussien et al. [8] concentrate on particular facts of healthcare data management. This method provides dynamic and detailed access control, but it may not scale well when there are many access policies, and it necessitates regular updates to the smart contract.

In summary, although these blockchain-based solutions present novel ways to tackle healthcare data management issues each has a unique set of drawbacks. Scalability issues, the need for performance optimization integration difficulties with the current healthcare IT infrastructure and the difficulties of striking a balance between off-chain and on-chain data storage are common problems. To validate these solutions' efficacy in a range of healthcare scenarios future research and development in this field must concentrate on addressing these limitations and conducting more thorough real-world testing.

3.5. Conclusion:

Notable developments in network architecture file storage document sharing access control and action logging are highlighted in this overview of blockchain technology in healthcare systems. Novel strategies that have shown enhanced data security privacy and scalability include the BSF-EHR framework [10] PatientDataChain [4] and IPFS integration [15]. Immuable action logging [2] and fine-grained access control mechanisms [13] further improve data protection and accountability in healthcare processes and frameworks like ACTION-EHR [11] and OmniPHR [14] highlight blockchains potential for safe and interoperable health record exchange.

Performance optimization and smooth integration with the current healthcare IT infrastructure continue to be obstacles despite these encouraging advancements. In the future studies should concentrate on developing standardized interfaces and APIs, improving hybrid approaches and conducting empirical analyses of blockchain systems in a range of healthcare scenarios. Blockchain technology has the potential to completely transform healthcare data management as

it develops. This could improve patient care and outcomes by increasing system security efficiency and interoperability. To fully realize the potential of blockchain in healthcare ongoing innovation and resolving existing constraints are essential.

3.6. Highlight:

The core of the paper, therefore, discusses what the authors think is a recommended Azure cloud-based solution for secure document sharing between patients and doctors, and how integrated Azure services used help in fully meeting the multifaceted healthcare document management requirements. The article does not contrast this suggested approach with blockchain possibilities. Instead, it stresses that there should be continuous development in this field, whereby eventual direction must be organizational necessities, technological competencies as well as legislations allowing for 'hybrid' arrangements capable of combining decentralized and centralized structures to create more protected and scalable healthcare document management system possible.

4. PROPOSED SOLUTION

4.1. INTRODUCTION

The healthcare systems are witnessing a notable advancement in data security privacy and interoperability through the application of blockchain technology. Nevertheless, several drawbacks are present in all implementations including difficulties with scalability performance bottlenecks integration with current IT infrastructure and the difficulties of striking a balance between off-chain and on-chain data storage. We offer a comprehensive solution that harnesses the strengths of several strategies while mitigating their individual shortcomings to overcome these constraints and realize the full potential of blockchain in the healthcare industry. This solution makes use of essential Azure services for a reliable and scalable implementation.

Our proposed solution aims to:

1. Improve scalability by using a dynamic multi-tiered storage architecture that can effectively handle huge amounts of healthcare data through globally distributed multi-model database management with Azure Cosmos DB.
2. With Azure Cosmos DB's high throughput and low latency access to data, we can enhance performance by strategically allocating data and computational loads.
3. Integration with existing healthcare IT systems will be facilitated using standardized interfaces and APIs via Azure Logic Apps for workflow automation and seamless system integration.
4. If balanced properly between on-chain and off-chain data management, Azure Cosmos DB can be used for secure off-chain storage of data and Azure Key Vault for managing keys safely and storing secrets to attain the highest security level and efficiency possible.
5. For instance, what is important to take note of is that Azure Storage is an elastic cloud storage solution geared towards unstructured data such as medical images and documents, among others related to health care systems.
6. This makes it very suitable for storing large amounts of healthcare data as well as ensuring efficient retrieval and management of those records especially for Azure Blob Storage which is part of the larger package referred to as Azure Storage.

4.2. SOFTWARE SPECIFICATION

In defining the proposed solution, we've focused on several problems:

- Restricted medication/drug selling
- Prescription procedure
- Medication adherence and monitoring
- Interoperability between healthcare providers and pharmacies
- Drug supply chain management

4.2.1. Problem analysis:

We've identified a number of scenarios that illustrate how our suggested solution can enhance the present workflows in order to better understand these problems in the context of the healthcare system. By digitizing paper prescriptions and enabling safe transfer to pharmacies Scenario 1 tackles the prescription procedure.

By allowing secure document sharing during consultations Scenario 2 enhances interoperability amongst healthcare providers.

Scenario 3 improves patient control over their medical records and the ability to share data. Many of the issues noted can be resolved by using these scenarios, especially those pertaining to patient privacy data sharing and prescription management.

4.3. SCENARIOS

The secure document sharing system is exemplified by the following scenarios, which highlight its key use cases.

Scenario 1:

This is a simplified scenario where the user's prescription is transferred to the pharmacy with the help of a secure document system. The doctor prepares a prescription on paper for the user at the start of the process. Next the user uploads an image of the paper prescription using the systems web application. You can accomplish this by either uploading an already-existing image file or taking a picture with their device's camera. After being captured the prescription is saved to the user's secure storage in the web application. The prescription will be securely stored in this vault accessible only to those who are authorized. The pharmacist has a pre-printed QR code that is intended for document sharing when the user visits the pharmacy to fill a prescription. This QR code makes the pharmacy identifiable and makes secure document transfers easier. The customer scans the QR code supplied by the pharmacy using the tablet or smartphone web application. The program helps the user choose the right prescription document from their safe storage while scanning. The user chooses which prescription they want to give the pharmacist access to. Following the user's selection of the prescription the document is securely transmitted to the pharmacy by the system. Data integrity and confidentiality are safeguarded during transmission through encryption. The pharmacy processes and dispenses the medication in accordance with the prescription after receiving it through its secure system.

Prescriptions can be transferred more easily from users to the pharmacy with the help of a secure document sharing system as this simplified scenario shows. By utilizing a web application safe storage QR code scanning and encrypted data transfer the system guarantees the secure and effective exchange of confidential medical data decreasing mistakes and enhancing user satisfaction.

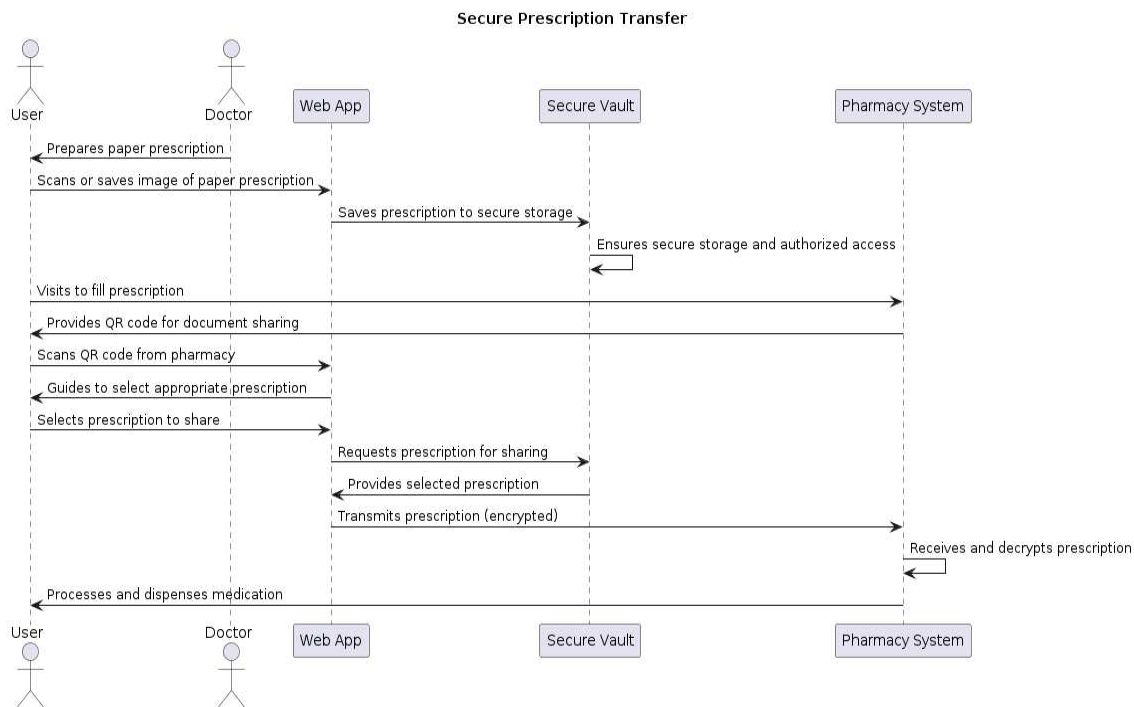


Figure 1) Sequence diagram for scenario 1

Scenario2:

In this instance sharing patient information with the doctor during the consultation is easier for both parties. A consultation with a specialist or other physician is scheduled by the user. If the patients' case-related data is required, the doctor will determine this during the consultation. From the healthcare provider the user receives this request. Via a secure channel that the user creates the healthcare provider can receive the necessary documents from the system. Among the options provided by the application are a temporary link secured by a PIN code and a matching QR code. Because temporary links have a finite lifespan and automatically expire, they guarantee that access to documents is restricted in time. By requiring the doctor to enter a PIN to view data PIN-protected links improve security.

Through device-scanning of QR codes the physician can obtain shared documents. Using the supplied URL PIN or QR code the physician can safely access the shared patient data. The healthcare professional utilizes the devices link provides the necessary PIN code upon request and makes use of the temporary or PIN-protected link. The shared documents are scanned by the physician using the QR code that is included with your device. Users have complete control over document sharing and they can use the web app to withdraw access privileges at any time. The temporary link will expire the PIN-protected link will stop working and the QR code will stop working if the user wishes to revoke access.

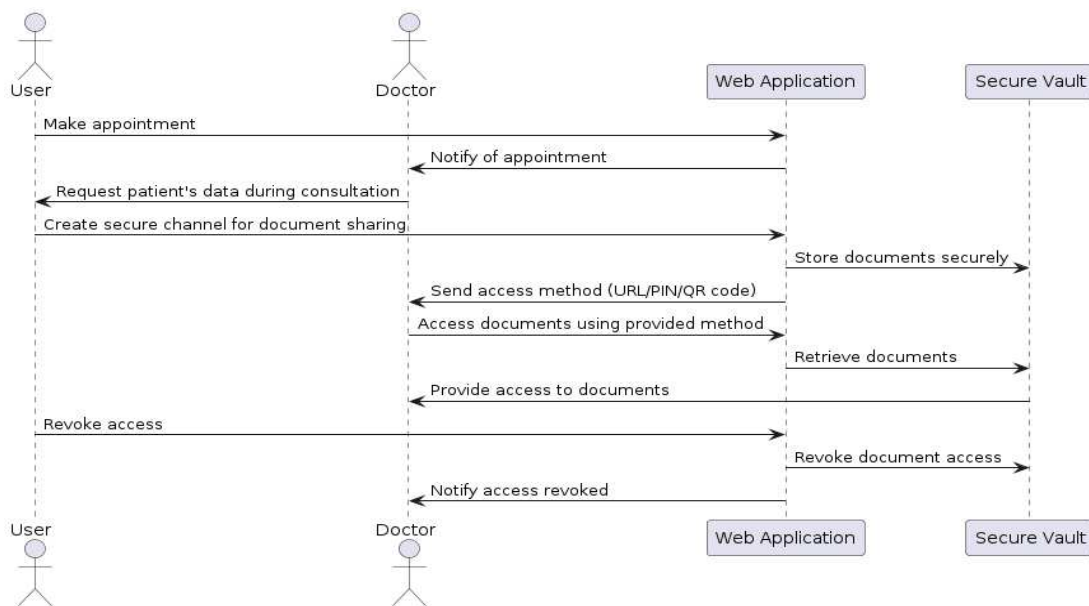


Figure 2) Sequence diagram for scenario 2

Scenario 3:

A document sharing system that is secure and online allows health care providers to ask for copies of medical records. On making such requests the providers get a notification about the specific details of the request. Requests can either be accepted or rejected by users. The web application used allows authorized persons to upload, share, and capture documents into their secure storage systems. All the links that are generated by this system are protected with a PIN code, QR code or a temporary link to doctor ensure safe sharing. Providers can use QR codes URLs or PINS in their web browsers to access shared documents. Users have control over who can view documents including the ability to revoke access and set time limits. All activities on any document in this repository are automatically logged by the system. The system allows specific

documents to be requested by healthcare practitioners and users may approve or disapprove any requests made upon them by anyone else. This web-based solution provides restricted, secure access only for authorized professionals that enhances efficiency reduces administrative workload streamlines medical data recovery processes.

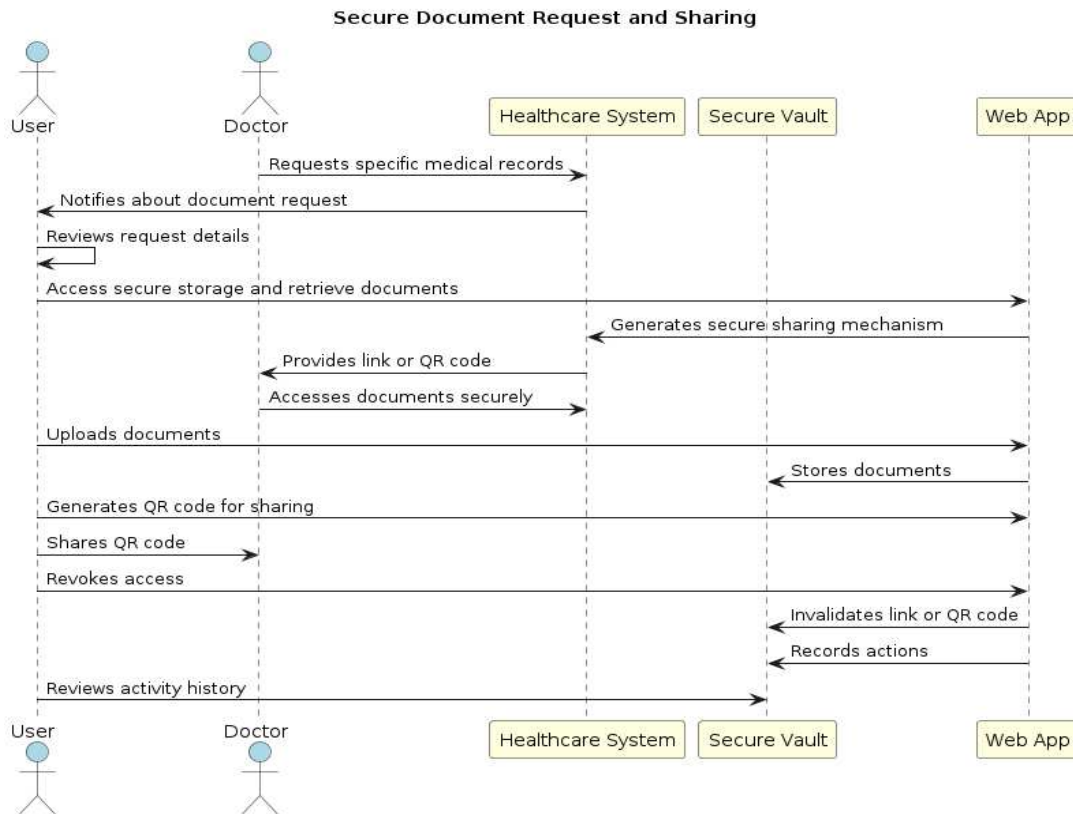


Figure 3) Sequence diagram for scenario 3

4.4. USER TYPES (ACTORS)

The following sections provide information about user categories in details:

User:

Access to a wide range of tools and features designed, to give users unparalleled control over their medical records while ensuring safe and efficient sharing across the healthcare system characterize this health record management system. The method adopts a user-centric design approach beginning with secure account creation and login processes that prevent unauthorized access to personal medical records. Users who successfully log in can store their medical records in a locked private vault. To satisfy different user preferences and situations the system offers a variety of flexible upload options. The ability to select files directly from devices allows users to upload pre-existing digital documents with ease. They are able to share files with other apps to facilitate seamless interaction with other digital health platforms or tools. A camera/scanner built into the system allows users to quickly digitize paper documents further bridging the gap between paper and digital health records. The system has robust and secure sharing features. Users can share their documents with doctors and pharmacists in a simple safe and secure manner by using QR codes. Rapid and secure patient data access is essential and this feature is particularly useful in clinical settings. It is essential that the system gives users a firm grasp of control. To safeguard their data ownership and privacy they can always decide who has access to shared documents and can withdraw access at any moment. To encourage transparency and accountability the system provides a comprehensive activity history log for each document. This feature allows users to monitor all document-related activities including who accessed what and when. The thorough audit trails improve security and give users a thorough understanding of the medical data lifecycle. The system also makes it possible for teamwork and efficient information sharing within the health network. Users can request documents from other users for instance in order to obtain medical records from a former healthcare provider. Yet they are able to accept and respond to requests for documents made by those with authorization. Enhancing the continuity and quality of care, this two-way data flow allows for the collection and sharing of all relevant medical records as needed. It also helps healthcare providers collaborate with one another thanks to the systems sharing request feature. This is particularly useful in situations where several healthcare providers must work together such as when a primary care physician needs to consult

a specialist. Through seamless collaboration, the technology promotes more thorough and well-coordinated patient care. The way that private health data is handled has advanced significantly with the help of this healthcare data management system. It allows users to manage their medical information and facilitates safe and efficient sharing within the healthcare ecosystem in addition to promoting enhanced provider collaboration. A more patient effective and secure management of data services is made possible by the system which addresses important issues like data privacy secure sharing and keeping thorough audit trails.

Pharmacy:

The users can share prescription documents with the pharmacy through its integration with a secure document-sharing platform. The pharmacy system safely receives prescription documents that a user chooses to share when they visit a pharmacy. The prescribed medications dosages patient data and any special instructions from the medical practitioner writing the prescription are all typically included in these records. Through the systems secure user interface pharmacists and authorized pharmacy staff can access these standard prescription documents. Pharmacists can confirm that prescriptions are authentic and accurate and that the right drugs are given to the right patients by looking through prescription records. Prescription document usage events are tracked by the system in a safe and auditable manner enhancing accountability and thwarting unwanted access.

The pharmacy system generates and shows pre-made QR codes for sharing the documents and information easier. You can access some prescription information, drug information and other pertinent data by scanning these unique QR codes. Throughout the pharmacy ready-to-print QR codes have been thoughtfully placed such as on the counter and on medication labels. These QR codes can be scanned by patients and medical professionals using the system Web app. The prescription information and other pertinent details from the medication history are automatically displayed when the QR code is scanned. This feature makes critical information quickly and easily accessible which enhances the accuracy and efficiency of the pharmacy workflow. Users can securely share prescription information with other approved parties like nurses or medical professionals using QR codes to facilitate seamless collaboration and continuity of care.

Healthcare Professional:

Medical professionals may request access to user data. This authorization ensures that they will be able to get the required medical information when they need it. Healthcare providers have access to shared documents through QR codes PINs or attached links. These access techniques allow for the simple retrieval of required patient information or prescriptions. Moreover healthcare providers may send requests for documents to users. This function allows them to directly request documents from patients or other users.

4.5. USE CASES PRESENTED IN DETAIL

4.5.1. INTRODUCTION:

The UML use case diagram in Figure 4 below depicts a prescription transfer system which allows safe sharing of medical prescriptions by patient's doctors to pharmacy. The purpose of this system is to ensure that the transmission of prescriptions is streamlined while maintaining the patients' privacy and data protection.

The diagram highlights several key actors and use-cases:

User: This is the main person who interacts with the system to control his or her prescriptions.

The primary prescriptions are written by doctors.

Pharmacy: They are responsible for receiving and processing the prescription.

Web application: It is a user interface that links users to the system.

A secure vault preserves prescription data.

Main use cases include:

Ready Prescription: Prescriptions written by doctors for the users

Users scan paper prescriptions, convert them into digital format.

Select Prescription: The user uploads the prescription they want it sent to the pharmacy.

Transmit Prescription: Then the user can select the data that he wants to share securely with the pharmacy via the portal.

Prescription Receiving and Processing: The pharmacy can see the selected file and deals with incoming medicines.

Figure 4 shows how these components work together to ensure smooth creation of prescriptions, right up until fulfillment, while keeping security as well as user convenience in focus along their paths (see Fig 4). It provides a high-level overview of what the system does as well as how different stakeholders relate regarding its operation in relation to transferring prescriptions within healthcare organizations. A use-case diagram represents a secure document sharing medical records management system where interactions occur between users, pharmacy staff and various functions across multiple systems involved in healthcare applications development process (see Fig 4). Thus, allowing individuals to manage store and share their health information securely while retaining ownership over their data [For instance, users can maintain control over their own health records] However pharmacists have more specific reasons for using the system, such as sharing documents needed for their work. This allows key information to be transferred among healthcare providers while still maintaining confidentiality and security in this environment. Additionally, it ensures that each user behaves correctly according to his or her role in the system.

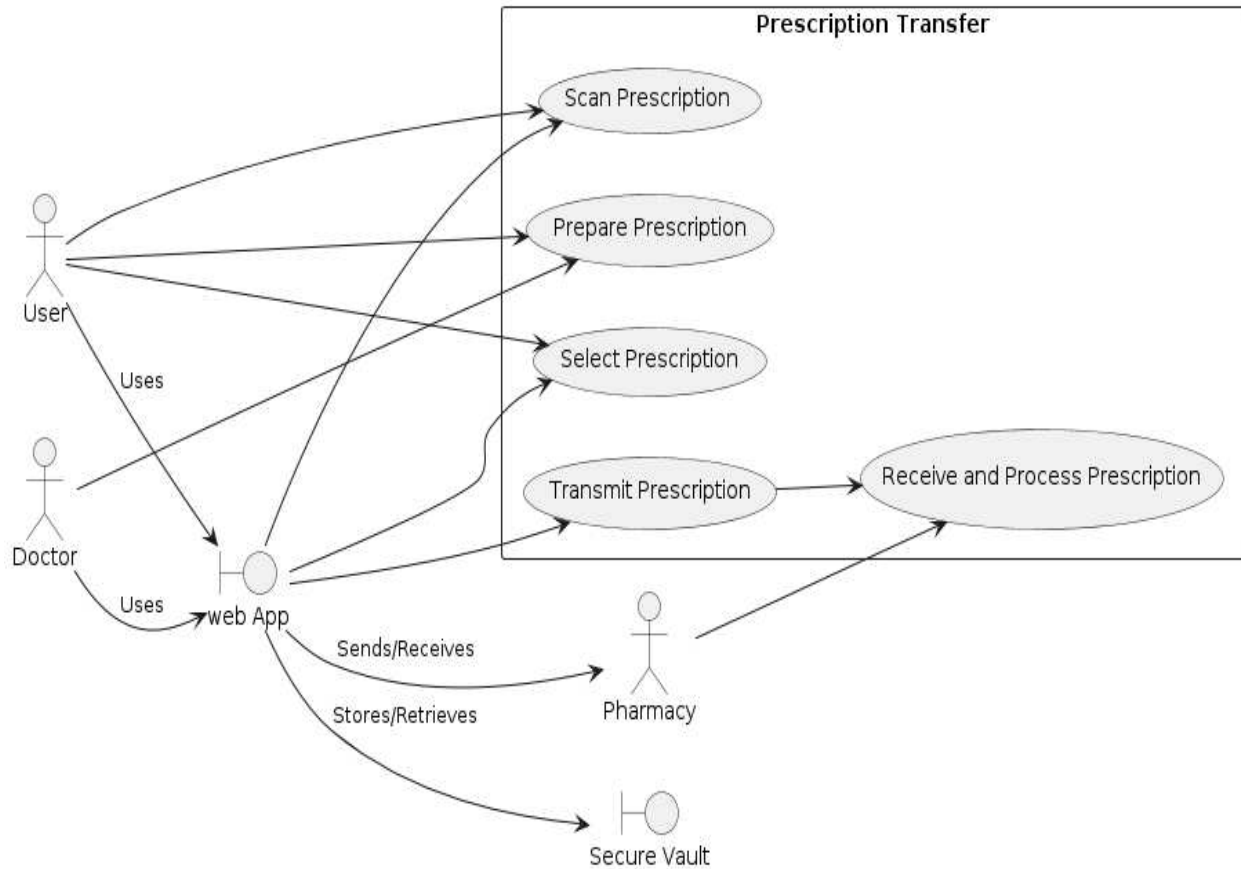


Figure 4) Use case diagram of the presented EHR system for secure document sharing

4.6 ALL-FUNCTIONALITIES USE CASE DIAGRAM

Functional Requirements:

1. User Sign-up and Verification:

- Enable users to sign-up and create safe accounts, log in with their credentials.
- Boost safety using multi-factor authentication.

2. Management of Prescriptions:

- Make it possible for doctors to write prescriptions.
- Enable users to scan or upload handwritten prescriptions.

- This will enable the users to see and control their own prescriptions as well.

3. Prescription Exchange:

- This ability lets users specify which prescription they want to process.
- This will assist in sending encrypted, secured prescriptions that can only be accessed by authorized individuals.

4. Integration of Pharmacy:

- Design of a portal where pharmacies allow digital or paperless prescription.
- The confirmation should be sent back by the pharmacy after receiving the prescription and the user will receive the confirmation email that the process is completed successfully.

Non-Functional Requirements:

1. Safety:

- Make sure that all transmissions of data should be end to end encryption.
- Update security protocols frequently to counter new threats as they emerge.

2. Performance:

- Ensure that all app functions are responsive.
- Many people should be able to access the system simultaneously without any performance problems.

3. Dependability:

- Have reliable measures for data restoration and error handling.

4. Scalability:

- Develop it to allow for number of the users and prescriptions.

5. Safekeeping:

- Prescriptions have to be kept secure, encrypted and store safely .

Establish access controls that make sure that certain prescriptions are stored and accessed only by authorized users.

Features of Web Apps:

- Have a user friendly user interface.
- Give the ability to choose, sent, deliver or save prescriptions.
- This is an infrastructure that can be easily expanded as the demand increases.

These specifications provide a comprehensive guide on designing a safe, efficient and easy to use prescription transfer system which satisfies exacting nonfunctional requirements of healthcare applications too.

4.7. INTERFACE MOCKUPS

This mock-up illustrates the MediDocs Electronic Health Record (EHR) system's user interface. This streamlined process demonstrates a secure and efficient method for sharing medical documents between healthcare providers and patients.

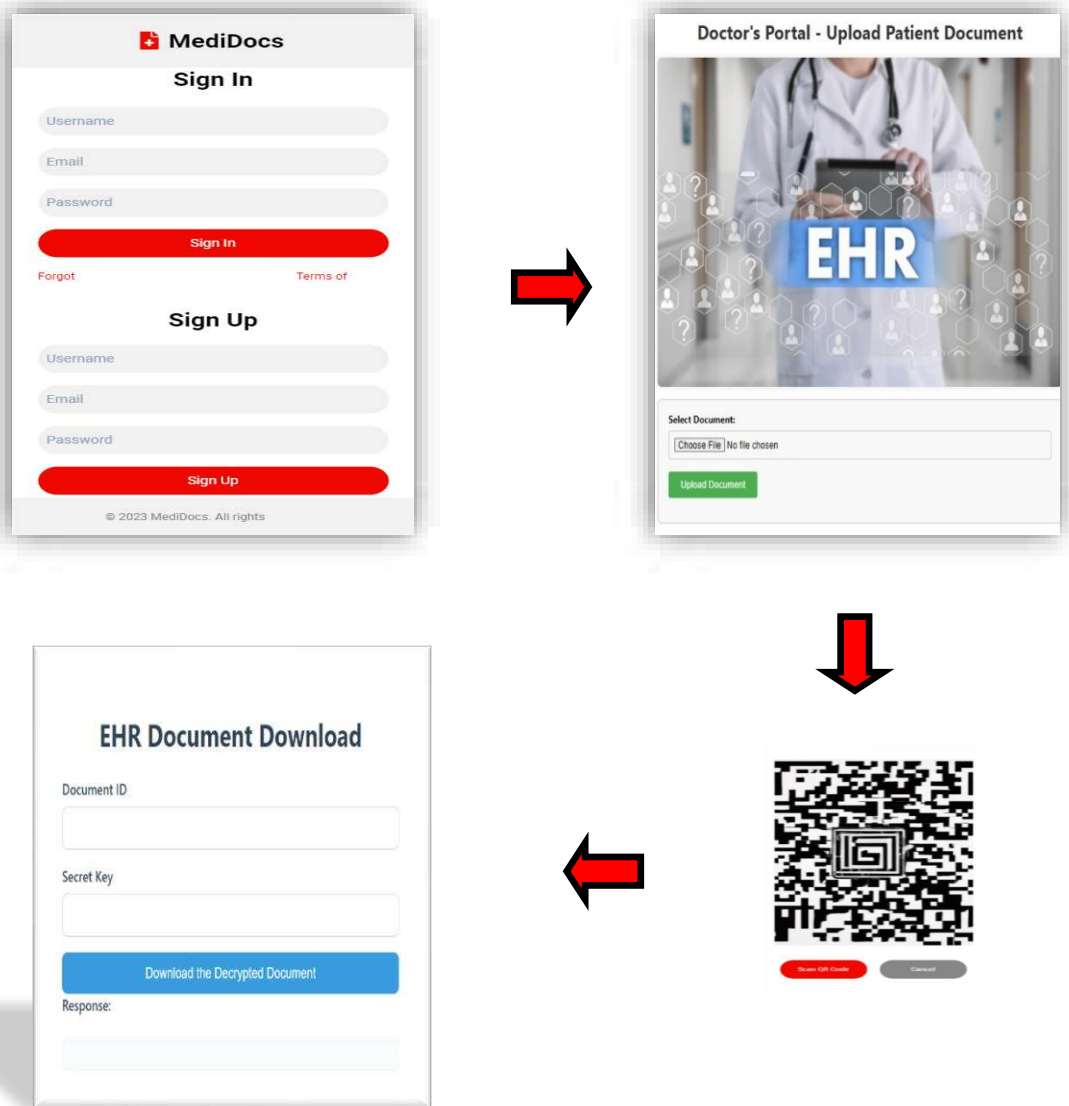


Figure 5) The Mockup of the proposed solution for secure document sharing

5. ARCHITECTURAL DESIGN

5.1 INTRODUCTION

The design of Electronic Health Record (EHR) systems is critical to ensuring effective safe and easily accessible patient care in the quickly changing field of healthcare technology. In order to meet the complex requirements of contemporary healthcare environments this section describes the architectural design of a cutting-edge electronic health record system that makes use of cloud computing. Built primarily on the Microsoft Azure platform, the proposed architecture integrates a variety of state-of-the-art components and services. In order to enable smooth document management, user authentication data storage and system integration this design attempts to build a reliable scalable and secure system. The architecture attempts to offer a complete solution that improves user experience while upholding the highest standards of data security and privacy by combining serverless computing cloud storage web applications and advanced database services. The high-level elements of this architecture will be covered in detail in the sections that follow along with an explanation of how each one affects the overall effectiveness and functionality of the EHR system.

5.2 HIGH-LEVEL COMPONENTS

The recommended solution for the given EHR (Electronic Health Record) system includes the following high-level elements:

Web application: An easy-to-use interface that makes sharing requests document management and system interaction possible. To orchestrate intricate workflows, it integrates with Azure Logic Apps and handles access control and document sharing queries.

Azure Cosmos DB: Documents metadata user data and activity logs are all stored in Azure Cosmos DB, a multi-model database service that is distributed globally. Its scalability and flexible schema make it perfect for efficiently managing a wide range of healthcare data.

Azure Key Vault: A safe service to store secrets and encryption keys needed to encrypt and decrypt private medical information.

The background processes of logic applications are directly connected to the web application, acting as the user interface. By integrating Azure Storage and Blob Storage, the architecture can also efficiently manage unstructured storage data such as stored images and documents. This configuration keeps the EHR system efficient, scalable, and secure. You must use a different method to handle user authorization and authentication.

Azure Logic Apps Workflows and processes can be automated with Azure Logic Apps, a serverless orchestration solution. It handles functions like data routing encryption and decryption of documents and notification triggering by integrating different system components.

Robust data management, secure key handling and effective workflow orchestration are the main objectives of this simplified architecture. Cosmos DB functions as the primary data storage solution providing scalability and flexibility for a wide range of EHR data. Azure Key Vault guarantees encryption key security which is essential for safeguarding private medical data. Coordinating operations between the application layer and data services Azure Logic Apps serve as the backbone of the system.

The Logic Apps backend processes are directly integrated with the web application which serves as the user interface. An effective scalable and safe EHR system is maintained by this configuration. Alternative methods would have to be used to handle user authorization and authentication, possibly within the application or via a third-party identity provider.

5.3 DEPLOYMENT AND COMPONENTS DIAGRAM

A thorough secure document sharing system for healthcare that makes use of Azure cloud services is depicted in the deployment and component diagram that follows. The users can securely upload medical records to the system and can then retrieve them. The users' Portal which allows for document uploads and the Pharmacy Portal which allows for document requests and retrieval make up the architectures two primary user interfaces. Several Azure services serve as the systems backbone and are interacted with by these portals. The Azure Logic App, which oversees document uploads encryption storage retrieval and notification procedures is the main orchestrator of this infrastructure.

Secure key management and encryption functions are handled by the Logic App through its interface with Azure Key Vault and the secure storage of encrypted documents and their related metadata is handled by Azure Cosmos DB, in addition the unstructured data is saved in the Azure blob storage, furthermore, To manage email or any patient notifications an external service is integrated. The data flow and interactions among these components are carefully delineated in the diagram which highlights the use of HTTPS/SSL protocols for all external communications to guarantee data security in transit. This architecture prioritizes data protection while facilitating effective information exchange between healthcare providers and patients. It shows how to manage sensitive medical documents in a cloud environment in a reliable scalable and highly secure way.

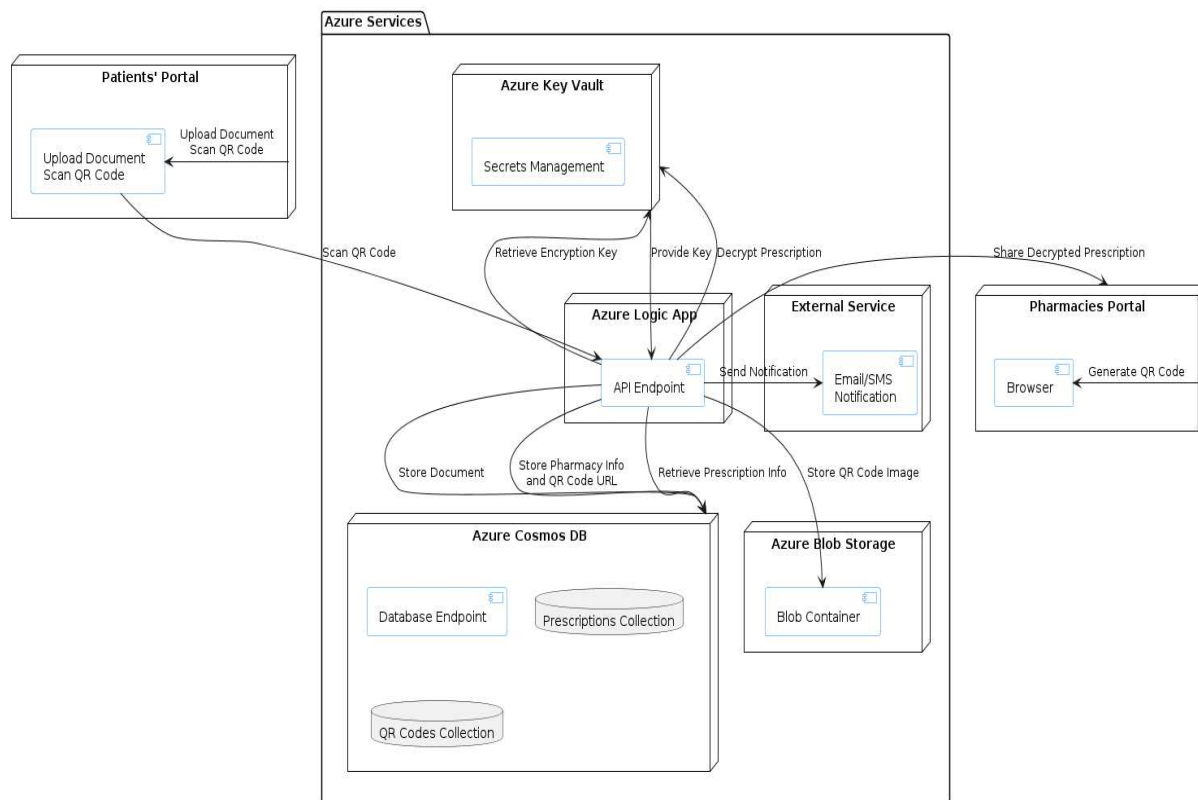


Figure 6) The deployment diagram for the secure document sharing

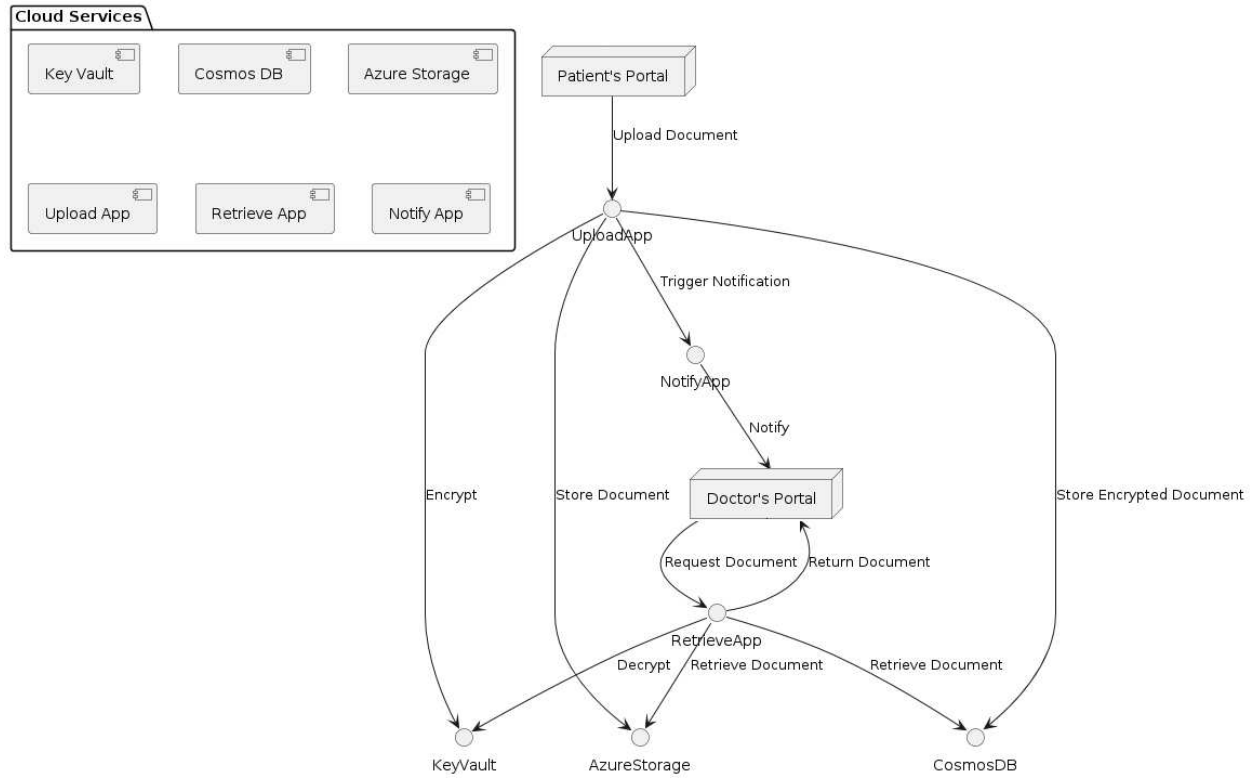


Figure 7) The proposed solution for secure document sharing component diagram

5.4 DATABASE STRUCTURE (ERD DIAGRAM + DETAILS) / FILE STORAGE

Below is an ERD diagram showing a prescription document in a safe healthcare data management system. The way this arrangement ensures that patients' privacy and security of sensitive health records are not violated is indicated by this. Characteristics of this data model include the following:

Document ID: Every document has a unique Document ID for easy management and retrieval purposes.

Patient Association: This makes it possible to have personalized health care by associating the prescription with a specific patient using the field Patient ID.

Encrypted Data: The Encrypted Data field contains key prescription details which show how committed the system is to safekeeping medical information. Encryption maintains confidentiality, while still being efficient about healthcare rules.

Metadata: Fields like `_rid`, `_self`, `_etag`, `_attachments`, and `_ts` give system level information that aids document management, versioning & tracking. This data structure achieves a balance between efficient data retrieval requirements, patient-centric organization and strict security measures. It should facilitate secure uploading of prescription documents into the system for storage as well as access without loss of integrity or compromise of private medical details.

ERD diagram's connection between Prescriptions and QrCodes entities further enhances functionality of Healthcare Management System enabling effective linking and tracking Prescription data with its associated QR codes. This approach integrates all aspects necessary for handling confidential health records effectively.

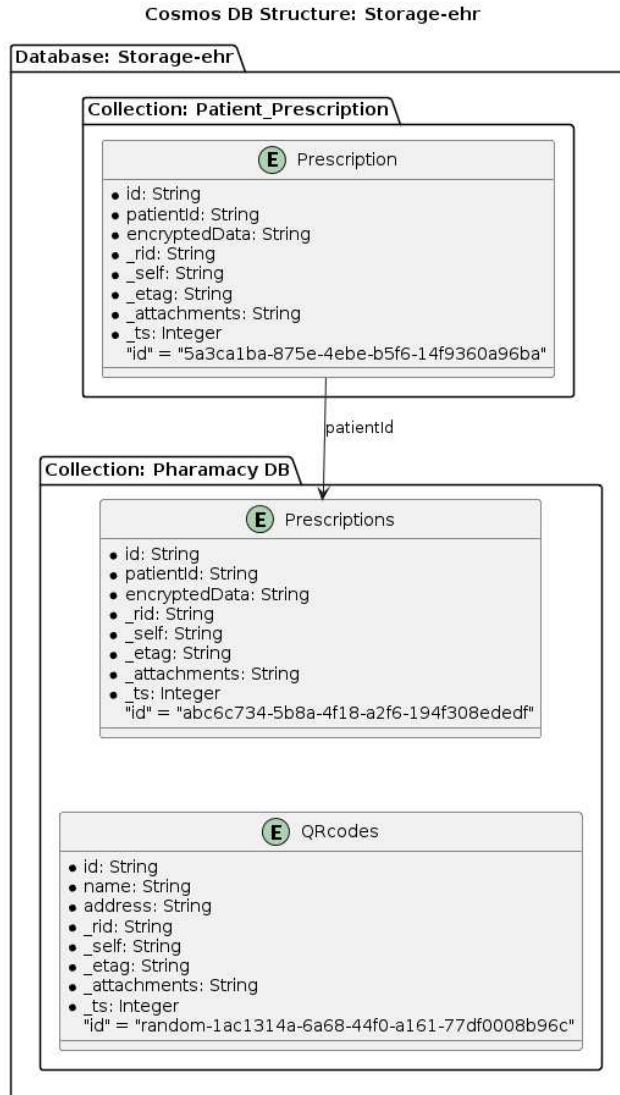


Figure 8) The ERD DIAGRAM of presented secure healthcare data management system.

5.5 SECURITY

Introduction:

The whole security plan of a cloud-based healthcare document sharing system is explained in this section. The major topics are: coordinated processes with Azure Logic Apps, encryption key management through the use of Azure Key Vault and secure document storage using Azure Cosmos DB. HTTPS with TLS 1.3 and EV SSL certificates are employed by the platform to

guarantee robust access control mechanisms as well as secure data transfer protocols. This multi-tiered approach ensures that there is a secure environment for confidential medical information thereby ensuring that all relevant aspects such as data integrity, availability, confidentiality and auditability are guaranteed throughout the lifecycle of the document.

To ensure document storage and encryption the system uses:

Azure Cosmos DB:

- offers globally distributed storage that is scalable for encrypted documents and metadata.
- Uses inbuilt encryption while in transit and at rest.
- Allows for disaster recovery and high availability across multiple regions.
- Before being stored documents are encrypted with the AES-256 algorithm.
- Every document has a unique encryption key generated by a CSPRNG.
- Each encryption operation uses Initialization Vectors (IVs) that are generated at random.

Azure Key Vault (AKV):

- It manages and secures encryption keys.
- Rotation of keys takes place regularly, for instance every 90 days to boost security.
- Before storing sensitive data like secrets in Key Vault, additional cryptographic techniques should be considered such as HMAC (Hash-based message authentication code) or PBKDF2 (Password-Based Key Derivation Function 2) in order to further enhance security with HMAC or PBKDF2 apart from SHA-256 hashing before storing secrets in the Key Vault which would add another layer of security to the secret information

Access Control:

- Access control logging and secure key management are offered by Azure Key Vault.

- Flexible control is provided by dynamic access decisions that are based on real-time data from Cosmos DB.
- Ease of management and consistency are ensured by the central storage of access rights in Cosmos DB.

The overall security benefits that the solution covers are:

Confidentiality:

- Secure key management and encryption are used to safeguard sensitive data.
- Integrity: Access controls and secure communication help to preserve the integrity of data.
- Availability: High system availability is guaranteed by Azure's infrastructure. Strengthening Compliance and traceability are possible through extensive logging and auditing capabilities.

Secure Data Transfer:

Data transfers between clients and Azure services are conducted over HTTPS with TLS 1.2 or 1.3. Azure services generally support contemporary cipher suites that put performance and security first. Applications that interact with customers can benefit from a high level of trust by using Extended Validation (EV) SSL certificates. Azure offers integrated encryption for information transferred between its datacenters. TLS 1.2 is the default protocol for all HTTP-based triggers and actions in Logic Apps. To secure data while it's in transit Azure Key Vault uses TLS. All data is automatically encrypted while in transit by Azure Cosmos DB. At the application level more security controls like these can be put in place:

- To lessen XSS attacks use Content Security Policy (CSP) headers.
- Limiting rates and throttling requests to lessen the effect of denial-of-service attacks.
- Malicious traffic can be filtered using the Web Application Firewall (WAF) feature of Azure Application Gateway.

6. IMPLEMENTATION DETAILS:

6.1 INTRODUCTION:

The safe and effective exchange of medical records between patients and healthcare providers is becoming more and more important in the quickly changing field of healthcare technology. This thesis addresses this need with a novel approach that makes use of Azure cloud services and prioritizes user-friendliness data security and patient privacy. Via an intuitive web portal, the proposed system allows the users to safely upload private medical records. To ensure the highest levels of data protection these documents are then processed, encrypted and stored using a combination of Azure services. Once their documents are ready patients can safely access them with a document ID and PIN.

This chapter will present the implementation details of a fully functional prototype. As presented in the architecture diagram of the proposed solution, there are three major sub-systems of the app; two web portals and a backend orchestrated with Azure Services including Key Vault for encryption, Cosmos DB for storing encrypted documents, Azure Storage for storing documents and various other Azure services for upload, retrieval and notification functionalities. The pharmacies' Portal and Patient's Portal comprise Web portals that interact with backend applications to facilitate key system functionality. The backend apps use Azure services to store securely, retrieve and inform users about their documents.

Further on, implementation details on each component will be presented.

6.2. Logic Apps

Logic Apps serve as a key service of our document sharing solution, orchestrating complex workflows, and managing critical processes. Three separate Logic Apps are implemented to handle distinct aspects of the system. To realize this scenario, we have used 6 logic apps inside the created resource group with the following functionality and configurations:

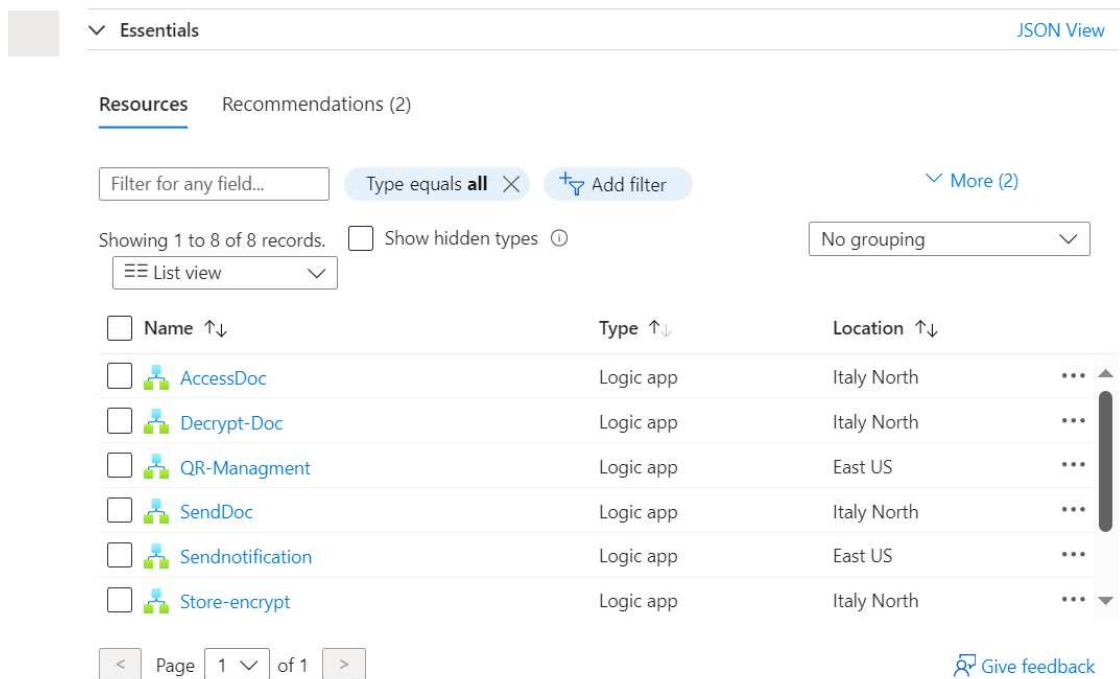


Figure 9) The created logic app inside the resource group

Store-encrypt:

This logic app receives the prescription document from the patient portal, where the document includes an encrypted PIN using SHA-256 hashing. The logic app then checks if there is a secret key in Azure Key Vault for this patient. In case it does not exist, it creates a new secret key in Key Vault, with the identity of an authorized Key Vault administrator.

The next step involves, employing this saved secret key to encrypt received prescription document. Finally, it saves the encrypted document into Cosmos DB's Patient Prescriptions container.

This process ensures the secure handling of medical information by:

- The use of SHA-256 hashing on PINs given through patient portals.
- Managing secret keys dynamically in Azure Key Vault with role-based access control mechanisms that help guard against any misuse of these keys.

Then it encrypts this prescription document using its secret key from key vault and stores it in Cosmos DB.

QR-Management:

This logic app is triggered as the pharmacy portal submits a request that includes pharmacy name and address in the message body. As an additional step, it saves to Azure Blob Storage a corresponding PNG file for the random QR code generated. Besides, it stores also in Cosmos DB the information about pharmacies connected by a document identifier. The connection between Cosmos DB and Blob Storage provides safe management of QR codes associated with prescription data.

AccessDoc:

This logic app is triggered if the patient portal sends an HTTP request with a pharmacy ID, a patient ID, and an SHA-256 hashed encrypted PIN. The first step in this process is comparing the pin provided by the user to the secret key saved in Azure Key Vault. It also validates that the pharmacy ID is valid for this organization After successful validation of the PIN, and Pharmacy ID then retrieves all prescription documents associated with that patient ID from Cosmos DB.

verifying the patient's identity using an encrypted PIN/Password as well as a pharmacy number. Make use of Azure Key Vault to securely retain encrypted secret key for encryption or decryption purposes

Send Doc:

This logic app will receive the document ID, pharmacy ID, and encrypted PIN (using SHA-256 hashing) from the patient portal. The provided PIN will be checked after which it will verify the provided pharmacy ID before retrieving an encrypted document associated with the chosen document ID from Cosmos DB Prescriptions container. After validating the given PIN as well as pharmacy ID, this logic app will go ahead to retrieve a secure document linked to selected document id from Cosmos DB `Prescriptions` container, then the logic app will then securely send the encrypted document to the pharmacy portal's database, ensuring the prescription information is protected throughout the process.

Here, the following security measures have been put in place:

- To authenticate and authorize this patient via encrypted pin and pharmacy ID validation.
- For privacy purposes, the encrypted document is retrieved from Cosmos DB.
- The encrypted document is then sent to the pharmacy portal's database

Decrypt-Doc:

This logic app will obtain the document ID, pharmacy ID, and encrypted PIN (using SHA-256 hashing) from the patient portal. Once confirmed that the PIN is correct and the pharmacy ID provided is valid, it will read out the encrypted document associated with this identifier from Cosmos DB `Prescriptions` container.

The logic app shall then decrypt the retrieved document using a secret key stored in Azure Key Vault. After being decrypted, this application must securely transmit prescription data to a pharmacy portal without breaching privacy or trust.

This workflow has the following security features:

- Validation of the pharmacy identity and authorization through an encrypted PIN and pharmacy ID check.

- The encrypted document should be retrieved from Cosmos DB 'Prescriptions' container for secured medical data handling purposes.
- The Azure Key Vault contains a secret key which can decrypt the document thereby ensuring that only authorized personnel can view prescription details.
- To prevent unauthorized access of information, decrypted prescription data should be sent securely to a Pharmacy Portal.

Through these steps, confidentiality and integrity of prescription information are maintained as it moves between patient portal and pharmacy portal. It also considers stakeholders' privacy and trust when sharing such information.

Send Notification:

Finally, upon successful transmission of prescription records to pharmacies; this application sends email confirmation to satisfy patients' concern about security transfer complete for their medical information.

6.3. Store-encrypt Logic APP:

The HTTPS trigger of this system's front-ends initiates requests to a logic app which is also in the system. In order to have secure communication, the request for the use of 256-bit SHA-256 hash algorithm and document message body is made. For illustration purposes only, user management functions such as user authentication and authorization are out of scope in this case..

The logic app, after parsing the received JSON message from the trigger, will compose only the received message not pin.

Interface the Azure Key Vault: The logic app as a vault administrator with an HTTP PUT request will either create or update a new secret in the patient's vault through the use of the PUT request using the inserted encrypted pin from the patient portal when upload the documents .

- To increase system security, the logic app will get the encryption key from Azure Key Vault connector and use it to encrypt the message body received from the patient portal.

- **Encrypt Document Data:** The logic app leverages the encryption capabilities of Azure Key Vault, specifically the RSA-OAEP-256 algorithm, to encrypt the contents or text of a prescription document using the retrieved encryption key."

- **Store Encrypted Data:** After encrypting prescription data, this logic app creates a unique document identifier, upon which it stores the following in an Azure cosmos DB collection:
 - **id:** The unique document ID
 - **patientId:** The associated patient ID
 - **encryptedData:** The encrypted prescription data
 - **rid, _self, _etag, _attachments, _ts:** Cosmos DB system metadata fields for efficient document management and tracking

- **Response Handling:** In response to the original HTTP(S) request made indicating successful processing and storage of prescription data.



Figure 10) The implemented workflow for securely storing the data.

QR-Management Logic APP:

- Pharmacy Portal Request: An HTTP(S) request received by the logic app from the front-end portal of a pharmacy with the pharmacy name and address on the message body is triggered.
- Generate Random Variables: Once again, two random variables are created within this logic app. One for QR code ID and another for document ID.
- Create QR Code: The QR code image in PNG format and is generated using random QR code id. The content of the QR code comprises of three fields namely, “name” and “address”, which are encoded within a UTF-8-character set.

- Store QR Code in Blob Storage: Generated QR code PNG file is saved by Logic App to Azure Blob Storage, where it utilizes randomly picked document ID as its filename.

- Store Pharmacy Information in Cosmos DB: Store data structure includes Pharmacy Name, Address and associated Random QR code string as document ID the more details provided as below;
 - **id:** The unique document ID
 - **name:** The name of the pharmacy
 - **address:** The address of the pharmacy
 - **encryptedData:** The encrypted prescription data
 - **rid, _self, _etag, _attachments, _ts:** Cosmos DB system metadata fields for efficient document management and tracking

- Response Handling: In response to the original HTTP(S) request, this logic app replies with a status 200 OK saying that processing has been successful and also that pharmacy information including QR code has been stored.

The technical capabilities demonstrated by this QR-Management logic app are as follows:

- Dynamic Variable Generation: This logic app generates random, unique IDs for the QR code and associated document, thereby ensuring data integrity and traceability.
- QR Code Generation: The Logic App creates a PNG image file using the provided pharmacy information that is encoded into the UTF-8-character set. It uses a library or service for generation of QR codes.

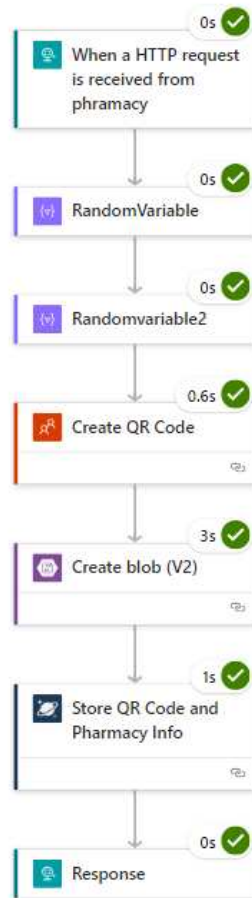


Figure 11) The implemented workflow for creating QR code.

AccessDoc Logic APP:

- Receive Patient Portal Request: The HTTP(S) request from the front-end portal of the patient that includes a pharmacy ID and a patient ID and the inserted encrypted pin in the message body triggers this logic app.
-
- To increase system security, the logic app will get the encryption key from Azure Key Vault connector and use it to encrypt the message body received from the patient portal
- Retrieve Pharmacy Information: For the given Pharmacy UD, this logic app looks up pharmacy details in Cosmos DB container by using physically scanned QR codes as document ID.

- Validate QR Code: If the QR code is valid, in other words, if it validates the information stored in the Cosmos DB Container then, this logic app can go to the next step and there is a condition that if the QR code is not matched then an error message will be shown and the portal doesn't direct to the next step and the process will terminated.
- Query the cosmos db using the patient ID : It returns all document ids related to provided patient id by querying cosmos DB container.
- Respond to Patient Portal: After query all document IDS using the provide patient Id , the logic app will respond to will confirmation message to the portal
- Error Handling: There are numerous error handling mechanisms built into the logic app to ensure that all exceptions or errors from any part of the workflow are dealt with gracefully, maintaining a reliable and consistent interface for the patient portal.

This logic app is designed to take advantage of Azure Cosmos DB's scalability capabilities in handling large volumes of prescription records It has been developed to provide fast response times even under load or growth constraints in terms of data or use. As a result, the following logic app implementation therefore offers the healthcare data management solution a secure way through which patients can access their prescription history all driven by document and patient ids. The technical design guarantees scalability, reliability, compliance with health regulations; thus, making it an effective component in the whole system.

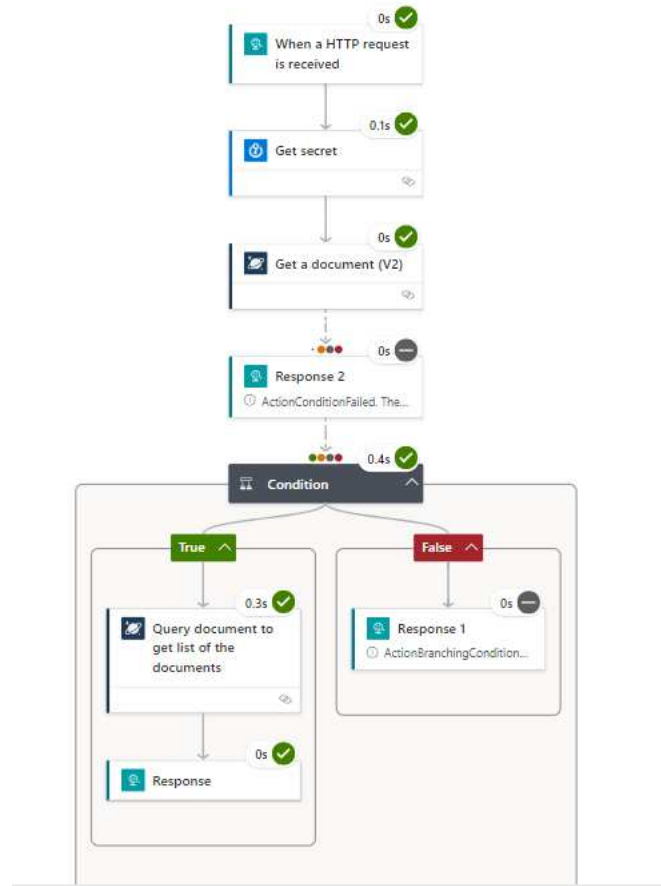


Figure 12) The document sending diagram

Send Doc App:

- Receive Patient Portal Request: The HTTP(S) request from the front-end portal of the patient that includes a pharmacy ID and a document ID and the inserted encrypted pin in the message body triggers this logic app
- To increase system security, the logic app will get the encryption key from Azure Key Vault connector and use it to encrypt the message body received from the patient portal.
- Retrieve Pharmacy Information: For the given Pharmacy UD, this logic app looks up pharmacy details in Cosmos DB container by using physically scanned QR codes as document ID.

- Validate QR Code: If the QR code is valid, in other words, if it validates the information stored in the Cosmos DB Container then, this logic app can go to the next step and there is a condition that if the QR code is not matched then an error message will be shown and the portal doesn't direct to the next step and the process will be terminated.
- Query the cosmos db using the document ID : It will return the document ids related to provided patient id by querying cosmos DB container.
- Create or update new document inside the pharmacy portal database using the provided document ID .

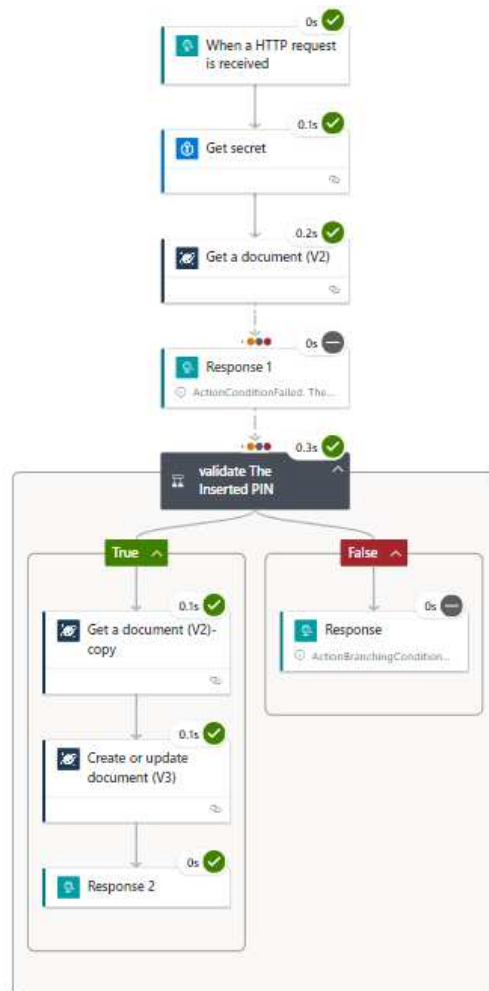


Figure 13) The implemented workflow for selecting the document ID and sending to the Pharmacy.

Decrypt-Doc APP:

- Receive Patient Portal Request: This logic app is triggered by an HTTP(S) request from the patient's frontend portal bearing the chosen document ID and the pharmacy id as its message body.
- To increase system security, the logic app will get the encryption key from Azure Key Vault connector and use it to encrypt the message body received from the patient portal
- Query Cosmos DB to Retrieve Encrypted Data: The logic app tries to query the encrypted data using a document id that it received.
- Get Pharmacy Details from Cosmos DB: At the same time, this logic app also uses a query which gives it access to pharmacy details.
- Compose Encrypted Data: Composing encrypted data retrieved from Cosmos DB
- Obtain Encryption Key from Azure Key Vault: This junctions with azure key vault thus accessing its secret key that was used in encrypting prescription data previously stored.
- Validate the inserted Pin with saved secret key , and if it is validate it will go ahead with process of the decryption
- Decrypt the encrypted Data: Using secret key obtained from Azure Key Vault, the cryptographic tool implements decryption using RSA-OAEP-256 algorithm on plain text information about prescriptions.
- Notification Triggering Logic App: On successful completion of decrypting prescription information, notify send notification Logic App with pharmacy ID as message body
- Patient Portal Response: Responds back to patient's front-end portal using original HTTP(S) request with status 200 OK indicating successful decryption and notification process with the decrypted data as a message body.

As result, Through this Decrypt-Doc logic app, healthcare data management solution can offer an efficient decrypted prescription record accessibility for patients and trigger notification process for

their respective pharmacies. In terms of scalability, reliability, and compliance with healthcare policies, this technical design becomes a crucial part of the whole system.

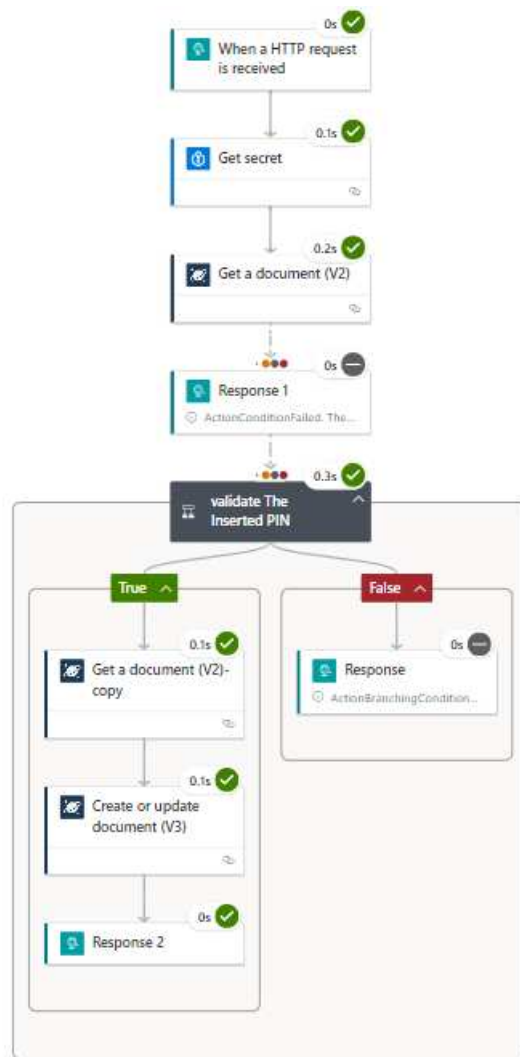


Figure 14) The implemented workflow for data decryption

Send Notification APP:

- Receive Trigger from Decrypt-Doc Logic App: This logic app is trigger when the document has successfully decrypted a document and passed to the pharmacy.
- Connect to Email Service: To send an email of notification to patients, a reliable email service provider such as SendGrid, Office 365 Outlook or custom email provider is linked with by this logical app.
- Sends emails to the patient with related pharmacy information.

Good Afternoon,

Dear Patient, Your prescription has been securely shared with Pharmacy Name: Pharmacy Address:

Thanks

Figure 15) The sample of the received confirmation email

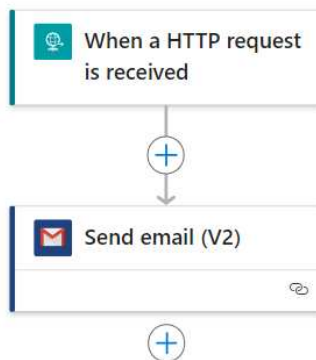


Figure 16) The implemented workflow for sending notification

6.4. User-friendly web portal for patients:

The solution gives patients an intuitive online portal designed just for them which makes it simpler for them to upload and share patients' private medical records. This portal serves as the primary interface for healthcare providers providing a seamless experience while upholding the highest standards of data security and privacy. So, they can easily upload patient medical records through the user-friendly secure online portal. By functioning as a graphical interface, it links healthcare providers to our solid back-end system which is supported by Azure services. It provides a simple way to select and upload different types of medical records such as prescriptions, test results and medical reports. Patients can also receive instant feedback on the upload process allowing them to know when their document has been successfully submitted. Through improved information sharing this improved method not only saves medical professionals valuable time but also raises the standard of overall patient care.

6.4.1. Technical Implementation:

To demonstrate the project, a simple HTML-based front-end was used at first to send EHR documents to Azure. This HTML way of doing things employed normal HTTP requests to interact with backend services, thereby providing a basic proof of concept for the document upload function. However, in real-life implementation of the EHR document upload system, a more advanced frontend architecture based on modern JavaScript frameworks can be adopted. In this way, it allows quite seamless and attractive user interface experience by employing the framework's ability for efficient data handling, real-time updates and smooth communication with backend services.

Integration with Azure Logic App:

- Handling of File Upload:

The EHR document upload system handles the process of file uploads asynchronously which makes it smooth and responsive to users. It is possible for a user to upload multiple files concurrently while the front-end application manages the concurrent file transfers seamlessly.

- Integration with Azure Logic App:

When a user initiates a document upload from web portal, an asynchronous HTTP POST request containing this document is created by front-end application that sends it to a predetermined endpoint.

- File Size and Type Restrictions:

To demonstrate and explain file upload functionality, a small text-only file has been used. This is not what will be done in the real environment where more advanced uploading types and size constraints would exist. The EHR document upload system would entail robust front-end application input validation with healthcare regulation adherence as well as effective data handling by limiting file types/sizes.

6.5. User-friendly web portal for Pharmacy:

A user-friendly online portal that is simple to use and safe for Pharmacy to access shared medical records between patients and the pharmacy. By providing privacy, security and accessibility this portal serves as a crucial part for patients and their medical records. Using this portal the pharmacy can insert the pharmacy information as name and address then they will receive a unique QR code that they can give to patient and patients can scan that Qr code then patients can select the desired document by insert the document Id from the list of available document Id and to share with pharmacy.

6.5.1 Technical Implementation:

For the initial proof of concept, a simple HTML-based front-end was used to send EHR documents to Azure. It relied on standard HTTP requests to the backend services in this basic HTML implementation which demonstrated how unique QR codes are generated. However, an advanced front-end will be employed for the actual real-world deployment of this EHR document upload system in order for the pharmacy to view decrypted documents by providing them with correct

The HTML implementation made use of “standard HTTP requests to interact with the backend.

These extra sections serve as more context and explanation that helps us see how we have moved from our first proof-of-concept towards actualizing a typical EHR document upload system.

More details ..

Integration with Azure Logic App:

The integration of the web portal with the Azure Logic App is explained in detail in the dedicated section covering the Logic App implementation.

The purpose of this online patient portal is to give people control over their health information. By offering a safe and intuitive user interface for document retrieval we enhance patient satisfaction and promote improved patient-provider communication. In addition to being convenient, document recovery is guaranteed to uphold the highest security and privacy standards thanks to the portal’s seamless integration with Azure Logic Apps.

Azure Cosmos DB:

For better performance and predictability, the capacity mode is set to provisioned throughput even though availability zones are turned off. Private endpoint is a connectivity method that provides secure, isolated connection to the Cosmos DB account. TLS 1.2 has been set up as the protocol for transport layer security; this ensures that there is encryption while communicating.

Periodically, data is backed up using the built-in backup and restore capabilities of Cosmos DB, and its data encryption uses service-managed key which Azure platform manages and maintains.

We have two collections of data: one for pharmacy and one for patient perception. The Cosmos DB database features robust security such as end-to-end encryption, role-based access control (RBAC), audit logging, besides global distribution capabilities for high availability and low latency access to health sensitive information.

The flexibility in adjusting to future scale requirements will be ensured by this application as well as changes in healthcare information management standards beyond just meeting today's need for secure document storage. Managed identities eliminate the need for storing or managing credentials between Logic Apps and Cosmos DB making it possible for secure keyless authentication enhancing system security further.

Azure Key Vault:

The Azure Key Vault we are using in our system has a specific configuration for the highest level of security it provides to all cryptographic operations and sensitive data handling. For fine-grained access controls to the Key Vault configuration, we have employed azure role-based access control (RBAC). RBAC permits us to make custom roles that have granular access controls, which therefore allows us to specify what users or applications can and cannot do in the vault. We initially set network access rules to "Allow public access for all networks" just for demonstration purposes but public access may be disabled while ensuring that a private endpoint network connection is used instead to further enhance security and restrict access only to authorized private networks.

The robust key and secret management features provided by Azure Key Vault are critical in maintaining comprehensive security of our document exchange solution as well as foster trust from patients and healthcare providers on safeguarding sensitive patient's medical information. On the other hand, Azure Key Vault delivers a scalable, flexible framework for managing cryptographic resources. This is achieved through safe storage of encryption keys securely within it which are then utilized during document encryption or decryption with logic app integration to protect patient data integrity as well as their confidentiality. It ensures that sensitive medical information is

handled safely and compliantly throughout the document processing workflow while enabling auditability or monitoring each instance where cryptographic assets are accessed.

7. RESULTS:

7.1 INTRODUCTION:

It is imperative to ensure the reliability, confidentiality and adherence of an electronic health record (EHR) management system because they contain sensitive medical data.

To deeply evaluate the performance, security, and general suitability for healthcare domain, it is highly likely that a variety of tests were conducted on this EHR management solution. As we focus on the major parts and join in a system, we may define several testing strategies that might have been employed to prove its behavior as well as making sure the system was ready for use in real life situations.

The sections below provide some insight about possible testing methods which include unit testing, integration testing checks applied towards evaluating the EHR management system described in the given information.

7.2 Test scenario:

7.2.1 Unit test:

This document provides us with the information that helps us to determine the following potential unit test scenarios that have been successfully executed:

- Front-end Implementation of Confirmation Message

- That front-end component has been tested extensively and confirmed as a success in showing confirmation message upon successful document upload.
- On the other hand, layout and UI elements were checked to ensure clarity of the message and user friendliness is guaranteed.
- Otherwise, there are meaningful error messages when an upload fails.

- Azure Key Vault Encryption

- It has been successfully proved that required encryption keys can be correctly retrieved by the logic app from Azure Key Vault.
- The encryption process was tested using sample data ensuring that encrypted output is formatted well for storage in Azure Cosmos DB.

- Notification Workflow

- This testing proves that after encrypting and storing a document, Logic App triggers notification workflow successfully.
- Integration with notification service (e.g., email, SMS) was verified to ensure message delivery as expected.

- Secret Key Validation

- The logic app's capability to validate patient's secret key during e-document retrieval stage was simulated.

- Document Decryption

- In addition, Decrypting stored documents by using validated secret key is another working test for this logic app.
- The decryption process passed the testing phase using encrypted data samples after which original documents were obtained back properly.

These are some examples of unit test scenarios that have been executed and the individual components or units making up the entire system have been verified to be working as expected. Successful execution of these unit tests ensures that we can rely on this behavior of the system as a basis for conducting subsequent integrations and end-to-end tests.

7.2.2 Integration test:

The information in the documentation has been based on some integration tests, which were carried out to verify interoperability and data flow between different components of a proposed solution. Thus, these integration test scenarios were completed successfully indicating that the system can act as one. Also, I have successfully done the subsequent potential integration test scenarios:

– Front-End Integration with Azure Logic App

- Extensive testing has been done for front-end interface and azure logic app integration.
- This involves confirming that the document upload request was successfully sent from the front-end to the Logic App with correct HTTP POST formatting and content.
- Through this process, it was ascertained that Logic App can retrieve encryption keys safely from Key Vault and then encrypt received data before storing encrypted document into Cosmos DB.
- The test cases have ensured that flow is end-to-end and sufficient data is passed to Logic App for other processing steps.

– Azure Logic App Integration with Azure Key Vault and Cosmos DB

- This involves successful testing of how Azure Logic App, Azure Key Vault, and Azure Cosmos DB integrate.
- These include ensuring seamless flow of data between various parts within EHR system through validating various points of interactions like data formatting and communication protocols among others.

- Notification Workflow Integration

- Having successfully completed the integration of notification workflows in a logic app and an outside notification service (email, SMS, etc.)
- Therefore, testing to see if notifications occur only after documents have been encrypted and stored (as confirmed by test cases) as well as ensuring that messages are delivered properly to their intended recipients.

These integration test scenarios have been executed without any glitch. This is why it has always been ensured by the development team through checking the integration points of the system so that all components fit well in each other giving a fully secure end-to-end solution.

The unit tests mentioned earlier combined with running these integration tests successfully make us sure how ready it is for further E2E testing and deployment.

As a result, the proposed solution is based on Azure services and may be compared with other blockchain-based healthcare systems discussed in the state-of-the-art:

- On a similar note, integration of Azure Key Vault for secure encryption and key management is just like advanced cryptographic techniques used by BSF-EHR model[10].
- like ACTION-HER[11] among others, where specialized health care scenarios are dealt with using Azure logic apps to orchestrate document upload, encryption and notification workflows.

7.2.3 Evaluation of the performance:

To achieve a critical analysis of the performance, it will be important to use available capabilities in the Azure cloud platform. This entails determining whether the system has sufficient capacity to handle increased demand for processing transactions and moving data. For example, the user can keep track of how Azure Services, Azure Cosmos DB and Azure Logic Apps are performing to determine if there is scalability provided across all these services that enable solutions on end-to-end scale up or down whenever necessary. This therefore ensures that the system can easily grow with expanding needs in the healthcare industry.

The evaluation of the transactions includes uploading, accessing and retrieving documents using our proposed solution. The performance of the various components such as the Azure Logic Apps and Azure Cosmos DB can help us gauge if we have such kind of transaction loads within an acceptable time thereby ensuring that our system has adequate capacity to process these transactions without compromising its performance. The overall latency can be evaluated by checking how long it takes for a document to be encrypted or decrypted from Azure Key Vault as well as when data coming from Azure Cosmos DB is sought after through its retrieval methods. By examining these statistics from azure in terms of their performance, we should be able to pinpoint any bottlenecks in this system that can help us optimize it and enhance better responsiveness for health practitioners and patients.

To effectively evaluate this proposed solution against stringent requirements like reliability, scalability and low-latency response time expected from healthcare systems; one must take advantage of advanced monitoring and optimization options offered by Microsoft Azure cloud service.

7.3. User story statement:

Whether you're a healthcare provider or patient, you will need an information system that is secured, dependable, flexible and responsive for uploading, retrieving and accessing electronic health records (EHRs) in order for me to effectively manage and distribute sensitive medical information to people authorized for it.

7.4. Security key points:

Securing Document Files:

Users need a framework that will safely store, retrieve, and access confidential EHR documents. The system should employ robust security measures such as encryption, access control and audit logs to ensure that medical data remains confidential and is not tampered with. It is important for users to know that they can have confidence in the safety of their data while uploading EHRs.

Dependability:

The Electronic Health Record management system must be reliable as it should suffer from a minimum downtime or performance problems. Healthcare providers rely on the system being available all the time and providing them with an easy way to manage patient records; thus, it is crucial that there are no any breaks or collapses in its operation. The system's architecture should be designed in such a way so it can withstand failures and bounce back when things do not go according to plan by ensuring high availability and durability of data.

Scalability:

For this reason, as healthcare expands requiring more volume of EHR data the system must scale accordingly. It should be able to accommodate large numbers of transactions like document uploads and retrievals without compromising overall performance. The system's architecture along with underlying infrastructure must be designed that expanding requirements of healthcare institutions and patients allowing smooth scaling up when it is necessary.

Minimal latency time:

Fast response times are expected by health care providers and patients when dealing with an electronic health record management system. To achieve this, Low-latency responses should be provided by the system so as not to make users wait for prolonged periods before they upload or get EHR documents. Quick retrieval of medical records enables physicians to make immediate decisions about how best to treat a patient.

7.5. Some feedback:**Speed and ease of development:**

The speed and ease of building the underlying infrastructure have been appreciated for the EHR management system. By using Azure cloud services such as Logic Apps, Cosmos DB and Key

Vault, the development team has been able to build quickly as well as deploy needed components without having to do lots of custom coding.

7.6. Key security highlights:

The security features of EHR management system have been a selling point to health practitioners and patients. Security-focused aspects have been mentioned regarding the proposed infrastructure:

Encryption:

Integration with Azure Key Vault for document encryption and decryption has provided a strong and secure way of protecting sensitive medical data. Users mentioned their confidence in the standard industry encryption algorithms and key management capabilities given by the provision of Key Vault service.

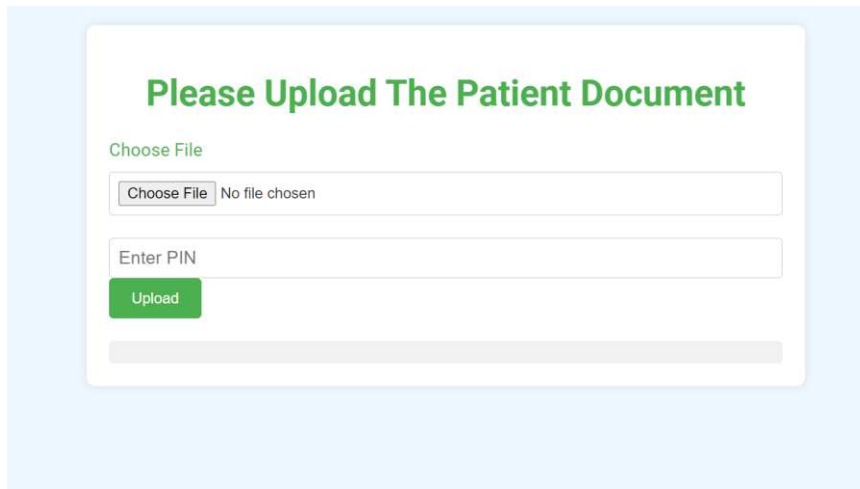
Access Control:

Through Azure Active Directory integration, fine-grained access control and role-based permissions have helped healthcare organizations to manage exactly who can access and interact with EHR documents. They agreed that the system can enforce strict access policies as well as audit user activities.

7.7. Front-end applications:

This part will explain the framework of the front-end that has been implemented:

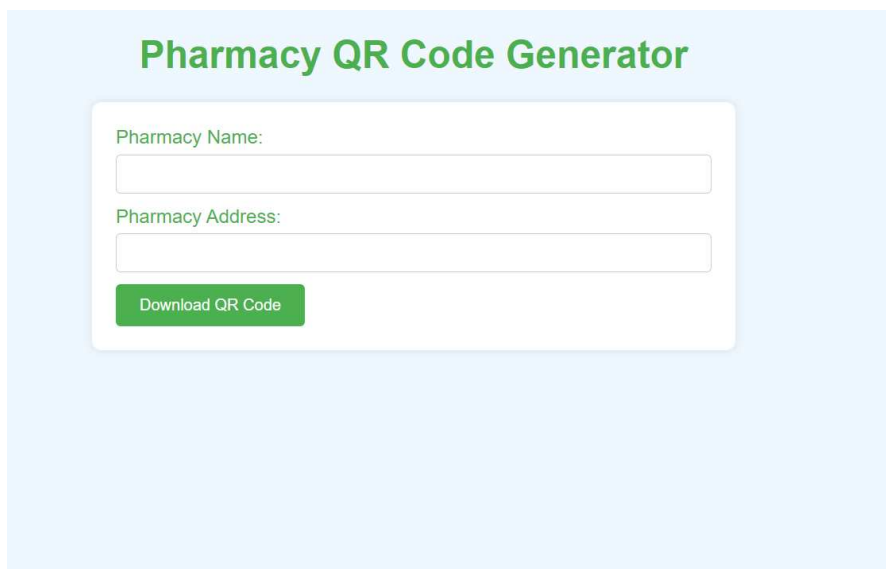
1) As the first step Patient uploads the scanned filed using this front-end interface and receive the successful message after uploading the document.



The screenshot shows a web form titled "Please Upload The Patient Document" in green text. Below the title, there is a section labeled "Choose File" with a text input field containing "Choose File" and "No file chosen". Below this is another text input field labeled "Enter PIN". A green "Upload" button is positioned below the PIN field. At the bottom of the form, there is a grey progress bar.

Figure 17) The patient uploads the documents with the PIN.

2) The pharmacy generates the unique QRcode and will download it .



The screenshot shows a web form titled "Pharmacy QR Code Generator" in green text. Below the title, there are two text input fields: "Pharmacy Name:" and "Pharmacy Address:". Below these fields is a green "Download QR Code" button.

Figure 18) The Pharmacy Insert the pharmacy Info to download the QR cod

3) The Patient get the list of the documents and can select each document Id to send to the Pharmacy with receiving the confirmation message.

Insert the Pharmacy ID, Patient ID, and PIN:

Pharmacy ID:

Patient ID:

PIN:

Select a Document ID:

Figure 19) The Patient gets the list of the document.

4) The pharmacy can see the information of the document ID and pharmacy in the portal

The image shows a web form with a light blue header containing the text "Dear Pharmacy, Please Insert the requested informations:". Below the header is a white form area with three input fields. The first field is labeled "Document ID:", the second "Pharmacy ID:", and the third "PIN:". Below these fields is a green button with the text "Receive the Document".

Figure 20) The pharmacy can easily and securely receive the document that patient has been sent.

The pharmacy can use this page to see the shared document that the user has already shared with them by inserting Document ID and Pharmacy ID and PIN.

All in all, with the presented minimal prototype we've demonstrated how can a Azure cloud-based architecture can be set up fast in order to accommodate specific scenarios, and ensure secure healthcare data sharing.

8. CONCLUSIONS:

The proposed Azure-based EHR management solution has promising possibilities for meeting core health IT system requirements such as reliability, privacy and compliance. Successful unit test execution and integration tests indicate that this system is behaving correctly and ready to be deployed in real life.

Extensive adoption of Azure services in the healthcare industry, and a wide range of cloud-based tools make it reasonable to assume that this decision was influenced by trends in the industry. The popularity of Azure services lies in its scalable infrastructure, strong security features that ensure patient confidentiality and data integrity, as well as an ability to seamlessly integrate with other systems.

These services also include Microsoft's Key Vault which provides secure encryption and key storage solutions while Logic Apps allows users to orchestrate document workflows within the same platform. Therefore, enhancing better development practices could save time spent on releasing new features or updates. To guarantee successful document upload requests with correct HTTP POST formatting and content, for example from a front-end interface to an Azure Logic App extensive testing has been performed. Furthermore, it has been confirmed that Logic App can request encryption keys from Key Vault before encrypting received data then saving the encrypted document in Cosmos DB.

Other test for integration between the three entities; namely Azure Logic App which should perform message routing based on content evaluation; Azure Key Vault where sensitive information should be stored securely; and finally Azure Cosmos DB that ensures smooth interaction among various components through appropriate data formatting are done successfully.

The set of azure services provides complete coverage hence allowing for enhanced security features, high scalability levels improved performance efficiency. Centralized nature is another benefit realized when using azure solution because it simplifies overall system management making future enhancements less resource-consuming which might take time otherwise. Alternatively, blockchain based architectures could provide entirely different approaches whereby documents are not kept on-chain but instead off chain while access permissions and hashes are

maintained on a distributed ledger. Consequently, it can enhance data integrity, patient control over medical records and transparency. Some of the challenges with blockchain solutions are related to transaction processing times, performance, scalability and the difficulty for development teams to acquire expertise.

7. References:

- [1] 1Password. (2023). 1Password Security Design. Retrieved June 30, 2024, from <https://1passwordstatic.com/files/security/1password-white-paper.pdf>
- [2] Ozdayi, M. S., Kantarcioglu, M. & Malin, B. Leveraging blockchain for immutable logging and querying across multiple sites. BMC Med. Genomics 13, 82 (2020).
- [3] Dukes, C. Blockchain Revolution: Ushering in A New Era Of Log Management
- [4] Cernian, A., Tiganoaia, B., Sacala, I., Pavel, A., & Iftemi, A. (2020). PatientDataChain: A Blockchain-Based approach to integrate personal health records. Sensors, 20(22), 6538.
- [5] Spil, Ton, and Richard Klein. 2014. Personal health records success: why Google Health failed and what does that mean for Microsoft HealthVault? In 47th Hawaii International Conference on System Sciences. IEEE.
- [6] Hailemichael MA, Marco-Ruiz L, Bellika JG. Privacy-preserving statistical query and processing on distributed OpenEHR data. Stud Health Technol Inform (2015) 210:766–70.10.3233/978-1-61499-512-8-766 - [DOI](#) - [PubMed](#)
- [7] Lee HA, Kung HH, Udayasankaran JG, Kijisanayotin B, Marcelo A, Chao LR, et al. . An architecture and management platform for blockchain-based personal health record exchange: development and usability study. J Med Internet Res. 2020. Jun 9;22(6):e16748. 10.2196/16748 - [DOI](#) - [PMC](#) - [PubMed](#)
- [8] Hussien H.M., Yasin S.M., Udzir N.I., Ninggal M.I.H. Blockchain-based access control scheme for secure shared personal health records over decentralised storage. Sensors. 2021;21:2462. doi: 10.3390/s21072462. - [DOI](#) - [PMC](#) - [PubMed](#)
- [9] ay Hales Hylock, Xiaoming Zeng A blockchain framework for patient-centered health records and exchange (HealthChain): evaluation and proof-of-Concept study." J. Med. Internet Res., 21.8 (2019), Article e13592
- [10] Abunadi, I.; Kumar, R.L. BSF-EHR: Blockchain security framework for electronic health records of patients. Sensors 2021, 21, 2865. [[Google Scholar](#)] [[CrossRef](#)] [[PubMed](#)]

- [11] Dubovitskaya, A.; Baig, F.; Xu, Z.; Shukla, R.; Zambani, P.S.; Swaminathan, A.; Jahangir, M.M.; Chowdhry, K.; Lachhani, R.; Idnani, N. ACTION-EHR: Patient-centric blockchain-based electronic health record data management for cancer care. *J. Med. Internet Res.* 2020, 22, e13598. [[Google Scholar](#)] [[CrossRef](#)]
- [12] Vanin, F.N.; Policarpo, L.M.; Righi, R.D.; Heck, S.M.; da Silva, V.F.; Goldim, J.; da Costa, C.A. A Blockchain-Based End-to-End Data Protection Model for Personal Health Records Sharing: A Fully Homomorphic Encryption Approach. *Sensors* 2023, 23, 14. [[Google Scholar](#)] [[CrossRef](#)]
- [13] Zhang, A.; Lin, X. Towards Secure and Privacy-Preserving Data Sharing in e-Health Systems via Consortium Blockchain. *J. Med. Syst.* 2018, 42, 1–18. [[Google Scholar](#)] [[CrossRef](#)]
- [14] Roehrs A, da Costa CA, Righi RR, et al. Integrating multiple blockchains to support distributed personal health records. *Health Informatics Journal* 2021; 27: 146045822110075.
- [15] Pilaes, I.C.A.; Azam, S.; Akbulut, S.; Jonkman, M.; Shanmugam, B. Addressing the Challenges of Electronic Health Records Using Blockchain and IPFS. *Sensors* 2022, 22, 4032. [[Google Scholar](#)] [[CrossRef](#)]
- [16] Harahap, N.C.; Handayani, P.W.; Hidayanto, A.N. Functionalities and issues in the implementation of personal health records: A systematic review. *J. Med. Internet Res.* 2020, 23, e26236. [[Google Scholar](#)] [[CrossRef](#)]