

PERFORMANCE EVALUATION AND ANALYSIS OF A CENTRALIZED PATIENT INFORMATION
SYSTEM FOR REAL-TIME DATA ACCESS AND CLINICAL DECISION SUPPORT

by

Osilonya Vincent¹, Isizoh A. N.², Ejeh Patrick³,

1,3: Department of Computer Science, Novena University, Ogume, Delta State, Nigeria

2: Department of Electronic and Computer Engineering, Nnamdi Azikiwe University, Awka, Nigeria

E-mail of the Corresponding Author: osilonyavincent@gmail.com

ABSTRACT

This paper discusses the performance evaluation and analysis of a centralized patient information system for real-time data access and clinical decision support. The paper actually evaluated the performance of the software which was developed using ASP. NET C# in Visual Studio for application layer, Microsoft SQL Server for backend data management. It offers role-based access controls, multi-factor authentication, real-time access to patient information, appointment booking, and full Electronic Health Record (EHR) support. The methodology used in the research was object-oriented analysis and design (OOAD) approach. Thus, the developed prototype was used to collect data samples, analyze how the users linked the system as well as iterating the process of system construction. The system's efficacy was tested in a sample of 10 variations of operation time (latency) to prove the point. The test findings also established an average lag of 127.5 milliseconds, proving the capability and agility of the system. In general, the system improves the delivery of health care in minimizing data redundancy, guaranteeing data integrity and empowering clinical decision making.

Keywords/Phrases: Centralized healthcare, data management system, object-oriented design, authentication, software.

I. INTRODUCTION

1.1 Centralized and Decentralized Healthcare Data Management

Centralized healthcare data management architecture" is a system where all patient data are stored on a single repository (database) and managed by unified database management system which is performed by a central entity (hospital, government agency, or cloud service provider) (Abughazalah et al., 2024). This amounts to efficient access to medical information which will significantly optimize the velocity of data access and the interoperability among healthcare services. Nevertheless, it comes

with its own challenges, including being a single point of weakness, more prone to cyber-attacks and lack of scalability when medical demands explode (Islam & Apu, 2024).

Based on the distributed ledger technology, Carta Healthcare (2024) proposed a centralized database system, as illustrated in Figure 1.

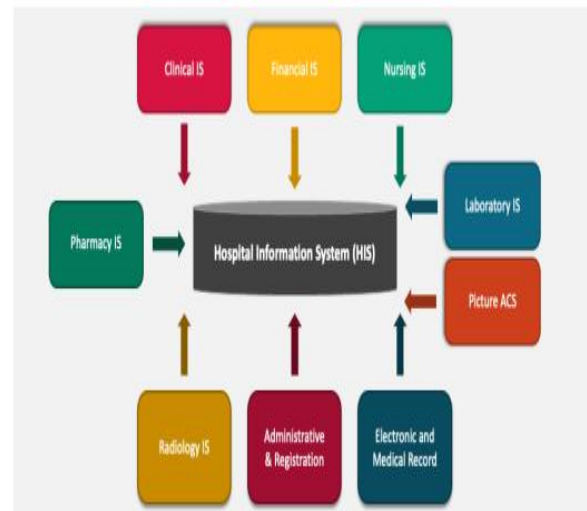


Figure 1: An example of a centralized database system for hospitals

1.2 Types of Electronic Health Record Systems

At present, EHR can be categorized into the following two types Centralized systems and Decentralized systems (Abughazalah et al., 2024): Centralized Systems: A central database is used for the purpose of storing the EHR of a hospital, or any healthcare organization.



Figure 2: Centralized data management system

Centralized systems have the benefit that they can provide improved data consistency, compliance with regulatory requirements, and data sharing among providers. They improve time-of-care decision-making, putting a patient's full record at doctors' and specialists' fingertips, for improved treatment. Controlled system centralization further allows a better security management, as security measures, technologies, and standards (e.g., encryption and authentication protocols) are centrally enforced throughout the system (Abughazalah et al., 2024).

However, centralized approach is also less suitable in some scenarios, as it has security risks, downtime risk, and lack of being controlled by individual health providers (Adeghe et al., 2024). In contrast, in decentralized healthcare data management, patient data is maintained across various locations and may be stored using blockchain or related technologies to ensure data integrity without relying on a single party (Yaqoob et al., 2022). This model enhances patient privacy by putting patients in charge of their records and reducing the potential for large-scale data breaches. Furthermore, decentralized systems improve fault tolerance; if one node fails, it will not impact the entire network (Veeramachaneni, 2025).

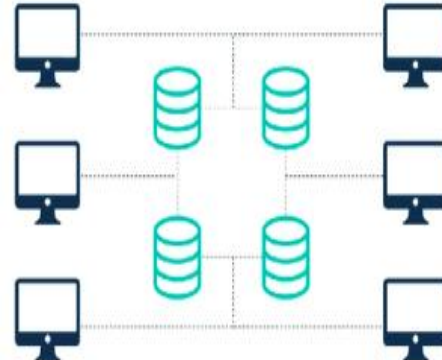


Figure 3: Decentralized data management system.

However, decentralized systems encounter challenges with interoperability, as the data formats and standards employed by various healthcare providers may differ, complicating information exchange (Islam & Apu, 2024).

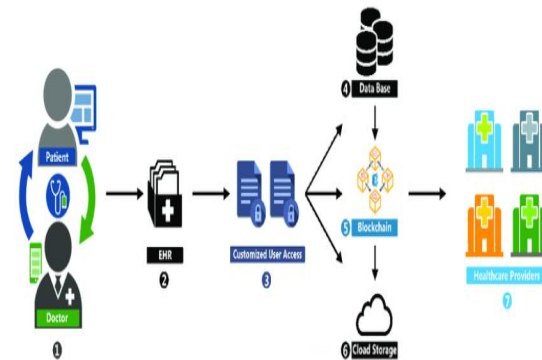


Figure 4: Healthcare Data Integrity Processing in Blockchain

A central and decentralized discussion exposes tradeoffs between efficiency and security. Centralized systems can have benefits on load balancing and faster data access and more standard protocols, but at the same time, the risks of single point of failure or data compromise are high. In contrast, decentralized systems are more secure in terms of patient autonomy, low reliance on infrastructural facilities with a loose level of control over the patient, but they require good protocols to ensure data consistency (Adeghe et al., 2024).

Two models are illustrated through the application of case studies to real-life situations. Health databases in Sweden and Canada are conducive to greater consistency between hospitals and medical specialists, and result in far more effective patient care. However, single point of failure comes with single points of catastrophic risk; for example, a

single cyber-attack (1.5 million patients' information were leaked in the health cyberattack in Singapore (Abughazalah et al., 2024).

Having said all these, the Estonian blockchain healthcare is an excellent one (actually it has a fantastic model that stretches 'patient own their record' to the next level and everything in blockchain and records are kept safely) In the end centralized healthcare data management vs decentralized data management, has to be solved by whom is more secure and whom is more accessible and whom faster can get in to a frame in a regulatory way. Traditional centralized models continue to be prevalent for their simplicity and easy implementation, decentralized models are gradually becoming more popular, especially in locations that focus on data privacy) and security from the cyber threats (Veeramachaneni, 2025).

II. LITERATURE REVIEW

2.1 Theoretical Frameworks

2.1.1 Clinical Decision Support System (CDSS)

Clinical Decision Support Systems (CDSSs) are medical computer programs developed to aid physicians in clinical decision-making by analyzing data from patients and suggesting recommendations. Such systems also maximize the accuracy of the diagnosis, the choice of treatment and the management of patients thus improving health outcomes (Elhaddad & Hamam, 2024). CDSS is important as it can decrease the frequency of medical errors, optimize workflow, and enable evidence-based medicine, and thus is a useful asset for contemporary health care systems (Michel et al., 2024).

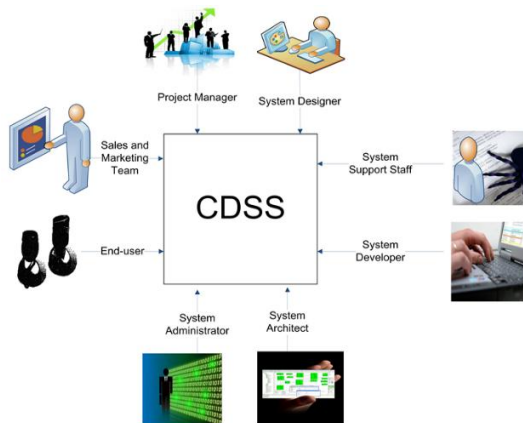


Figure 5: CDSS Provides Active Diagnosis Services to Clinicians

2.1.2 Intervention Phases for Clinical Decision Support Tools

Clinical Decision Support System (CDSS) can be classified into various groups according to their functions:

- Therapy Alert and Reminder: Such alerts and reminders inform healthcare providers about potential drug interactions, allergens, and missed screenings; thus, they can assist in adhering to best practices (Michel et al., 2024).
- Predictive Analytics: CDSS utilizes statistical models and AI algorithms to predict disease risks, treatment outcomes, and patient deterioration, providing early warnings (Lin et al., 2024).

Recommendations and Guidelines: Some CDSS systems offer customized medication recommendations using patient records, medical history, and established scientific protocols (Chen et al., 2014).

2.1.3 Security and Privacy in Health Data Management

Centralized healthcare systems have enormous amounts of highly sensitive patient information and, as such, are primary targets for cyberattacks. Among the most serious risks:

- Data Breach and Unauthorized Access: Hackers use weak authentication schemes to access EHRs and other sensitive data (Shojaei et al., 2024).
- Ransomware Attacks: Computer malware encrypts health data, and a "ransom" must be paid to decrypt it, usually disrupting essential medical practices (Bala et al., 2024).
- Insider Threat: The illegal access to patient data by healthcare providers and third-party data processors is a major threat to patient privacy (Adeniyi et al., 2024).
- Insecure Data Transmission: Poor encrypting of data sent when transmitting information between hospitals, labs and policyholders results in data being left prone to cyberattacks (Khatiwada et al., 2024).
- Integration of Two-Factor Authentication (2FA) and Encryption

To reduce the risk of these threats, healthcare organizations have developed advanced security technologies such as:

- a. 2FA: Introducing an additional step to authenticate a user without relying on their password (e.g., fingerprint reader, OTPs) improves access control and decreases illegitimate logins (Shojaei et al., 2024).
 - b. End-to-End Encryption: Encryption secures the storage and communication of patient data, allowing only trusted users with the decryption keys to read or modify the sensitive data (Bala et al., 2024).
 - c. Security on the Blockchain: DLT facilities This tamper-proof audit records prevents data tampering, created trust medical records and transactions (Adeghe et al., 2024).
- ii. Privacy Regulations (e.g. HIPAA, GDPR, etc.)

Regulatory compliance is of utmost importance for protecting healthcare data:

- a. Health Insurance Portability and Accountability Act (HIPAA): It is a law that provides privacy and security standards to protect medical information in the United States of America with provisions regarding (Adeniyi et al., 2024).
 - b. General Data Protection Regulation (GDPR): It outlines the specific requirements of data capturing, usage and consent processes of medical organizations in Europe (Popoola et al., 2024).
 - c. Data Sovereignty Laws: They are country laws made to bring about governance that govern the flow of patient data and the way end up with shared outside the national boundaries preventing unauthorized sharing of these data and the countries think it is as a form of risk control (Yang et al., 2024).
- iii. Approaches for Improving Security and Privacy of Data

In order to enhance the level of security in healthcare data, several measures are taken:

- a. Federated Learning: A machine learning paradigm that allows hospitals to train AI models from the distributed data on patients without revealing their privacy (Yang et al., 2024).

- b. ZTA: With the practice of constant identity monitoring and the policy of least privilege, the possibility of insider threats is minimized (Shojaei et al., 2024).
- c. Anomaly Detection Powered by AI: Real-time ML-powered Cyber threat discovery and containment minimizes probability of ransom ware and unauthorized access (Bala et al., 2024).
- d. Improved Patient Consent Mechanisms: Consent management systems on the blockchain facilitate the management of patient consent regarding who can access their data and for what purpose (Adeghe et al., 2024).

2.2 Review of Past Related Research Works

Abell et al. (2023) investigated barriers and facilitators to successfully implementing Computerized Clinical Decision Support Systems (CDSSs) in hospitals. Using the NASSS framework, their study revealed that barriers included technical implementation challenges, user resistance, and insufficient data interoperability, while facilitators included organizational support and training. Yet their research did not provide a comprehensive examination of the privacy and security concerns associated to the adoption of MHealth, in particular with respect to more sensitive decisions, such as the management of patient's health data in a more contemporary era. More importantly, it did not introduce any AI solutions to improve the efficiency of CDSS and provided no AI-based direction to enhance the effectiveness of CDSS.

Ademola et al. (2024) focused on the interoperability of EHRs. They presented a technical and semantic interoperability architecture for EHRs that would enable more efficient sharing of information across healthcare systems while protecting privacy and security. The study highlighted the necessity of standardized data transfer protocols to avoid data siloing and ensure effective patient information exchange. However, it was lacking in practical deployment and testing of the proposed interoperability solutions. It also failed to examine the deployment scalability of the proposed methodology in extensive healthcare networks.

Chaparro et al. (2022) conducted research on clinical decision support stewardship, focusing on best practices and methods to track and enhance

interruptive alerts. They examined the impact of interruptive alerts in CDSS on clinical decision-making, and described strategies for reducing alert fatigue, improving alert quality, and maximizing the effectiveness of decision support interventions. However, this study did not explore the use of machine learning to enhance the accuracy of alerts or to personalize decision support for specific patients. Additionally, it did not address the potential dangers of cybersecurity attacks on the integrity of CDSS.

Chen et al. (2024) developed and instituted an Electronic Health Record (EHR)-embedded Clinical Decision Support System (CDSS) trial. In this systematic review, we evaluated EHR-integrated CDSS trials in traumatic brain injury (TBI), demonstrating their efficacy in enhancing the accuracy of diagnosis and treatment recommendations. The report also highlighted the challenges encountered: issues with adoption rates and system sluggishness. Nonetheless, this study did not consider the combination of blockchain with federated learning to improve data security and decentralized decision-making in CDSS trials.

Jancy et al. (2024) presented the DigiMedi System, a theoretical model of a web-based health information system aimed at reducing patient record accessibility and streamlining hospital workflows. However, the approach was limited to the framework, with no further implementation details or real-time performance presented. It also did not account for tools such as homomorphic encryption, a privacy-preserving technology that could be used to protect patient records.

Karaferis et al. (2025) proposed a data-driven AI system to minimize variations in healthcare EHR usage. Their research introduced an AI-based strategy to enhance EHR effectiveness and maintain the required accuracy for processing healthcare data, primarily focusing on predictive analytics and intelligent automation. However, it did not address the ethical and regulatory issues related to EHR processing using AI. Additionally, concerns about data privacy risks in adopting AI in healthcare were not thoroughly discussed.

Michel et al. (2024) performed a scoping review of CDDs in emergency telephone triage, exploring their design, development, and use. They aimed to optimize response times and reduce triage errors. However, it did not examine how real-time data

analysis and AI-based diagnostics could further streamline the triage process.

Mortier et al. (2024) developed a CDSS prototype (PERMANENS Project) to assist staff in the emergency department with managing self-harm cases, connecting with mental health risk assessment tools. However, their research did not reflect on how modern blockchain technology could help maintain patients' privacy concerning mental health cases.

Tilala et al. (2024) investigated how healthcare data lakes enable real-time processing of data for immediate decision-making and patient care interventions. Although it addressed real-time data processing, it did not consider how AI-based predictive analytics might enhance patient care.

III. RESEARCH METHODOLOGY

3.1 Methodology Adopted

For this research, an Object-Oriented Design (OOD) approach has been chosen and applied to support a structured modular development of the system. This method was preferred over alternatives such as DevOps because it emphasizes clean architectural design, code reusability, and long-term maintainability, important qualities for healthcare information systems with high demands on reliability and security. OOD facilitated the definition of system components with encapsulated functionality, making them easier to test, debug, and modify as the system evolves.

The development process was structured according to OOD principles, encompassing analysis, design, implementation, and testing. As part of systems analysis, clinical workflow and data needs were reviewed in detail to determine the central objects and their interrelationships. During the design phase, the class hierarchy of the main components of the system, such as patient records, user authentication, and clinical decision modules, was defined. Microsoft.AspNet.Identity (on the authentication side) was integrated into the class hierarchy, but instead of using AES encryption, it employed secure hashing of patient credentials.

Implementation leveraged ASP.NET (C#) as a frontend framework and utilizes HTML/CSS/JS for UI development. The project achieved quick, responsive data interactions by using jQuery combined with AJAX while maintaining a clear separation between interface components and business logic. The backend successfully utilized

Microsoft SQL Server with data access layers constructed from discrete objects based on OOD. This object-oriented architecture provided a flexible design for query optimization and secure data handling.

System testing adopted a more integrated perspective, assessing both functionality and performance. Response Time (latency) was tracked as a primary metric for real-time data retrieval, focusing on optimizing object interaction patterns to ensure sub-second response times for clinically critical functions. Security testing confirmed that role-based access control (RBAC) and multi-factor authentication (MFA) had been properly implemented via the Microsoft.AspNet.Identity framework.

3.2 The Analysis of the New System

The centerpiece system is the CPIS (centralized Patient Information System) as a comprehensive end-to-end solution for the lack of integration of existing fragmented healthcare information systems. Through Object-Oriented Design, the system implements an integrated architecture for collating patient information into a secure central database and providing real-time access to identifying healthcare providers within the system. With data silos eradicated and information formats standardized across departments and locations, an exhaustive patient history can be drawn at the moment of care.

Real-time decision support is where the system truly shines. Other features include embedded algorithms and automatic alerts to provide healthcare professionals with instant, actionable information at the point of care about such aspects as lab results, treatment histories, and evidence-based guidelines. This capability greatly minimizes diagnostic lag time and improves the accuracy of treatment. The CPIS has seamless interoperability based on the HL7/FHIR standard, efficiently sending and receiving data among different healthcare providers and is

accessible on a cross-platform basis, thanks to web-based ergonomic interfaces.

Security is a fundamental requirement of the system architecture. The CPIS offers a robust defense with RABC (Role-Based Access Control), MFA (Multi-Factor Authentication), and full data encryption. These measures enable the protection of sensitive patient information while keeping it accessible enough for healthcare providers. Performance has been prioritized during development, optimized for sub-second response times for key operations and constructed to scale effectively with an increasing patient load.

3.3 Data Flow Diagram of the Proposed System

Figure 6 depicts the Data Flow Diagram (DFD) of the centralized Patient Information System, which describes how the data moves around the patient system. The system is immune to users (e.g., doctor, patient, administrator) who can only access this technically secured system after proper authentication. Patient data are entered, verified and retrieved on-line in real-time to allow prompt diagnosis, new therapy and reporting. Every patient history, prescription, and lab is tracked in one central database, shielded with best-in-class encryption and role-based access controls. The software also supports, in an automated fashion, the creation of reports, storage of digital maps, and data visualization, thus promoting accuracy, rapidity and best clinical management.

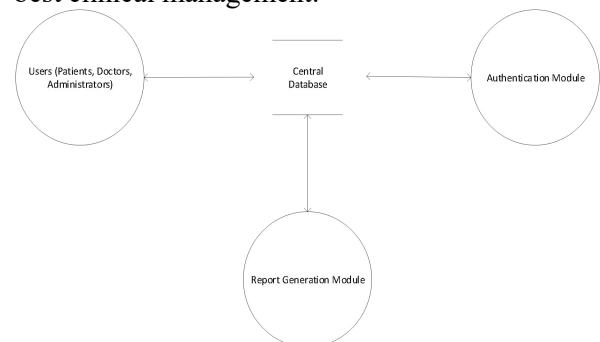


Figure 6: Data flow diagram of the proposed system

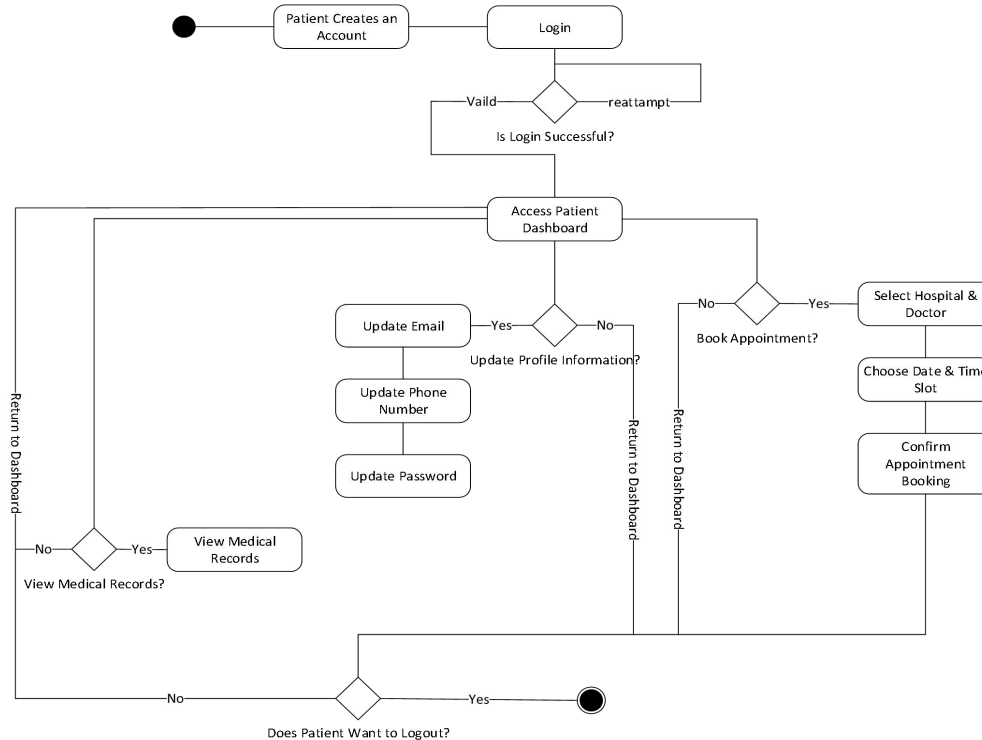


Figure 7: Patient Activity Diagram

3.4 Entity-Relationship Diagram (ERD) of New System

Dependencies among core system entities are shown in Figure 8 to maintain the proper flow of patient data, as well as hospital administration and clinical workflow. The system conforms to the standardized Role-Based Access Control (RBAC) pattern and supports Multi-Factor Authentication (MFA), including both password and one-time passcode (OTP), to provide a higher level of security.

The system is controlled at various levels. It's a hierarchical structure, at the highest level being the Super Admin (MOH), and then at the next level Sub Admins (Hospital management), whose jobs is to manage the respective hospital accordingly. Each hospital is composed of a variety of users, which could be doctors, nurses, lab technicians, and patients, and interacts with the system by playing the role of an individual.

Doctors can see patient records, add diagnoses and medications, and also order lab tests. They schedule

appointments and form a many-to-many relationship with patients. Nurses help treat, monitor and update and care for the patients. Lab technicians administer diagnostic tests, manage the transmission of test results, and transfer reports that help doctors and patients interact in a many-to-many relationship. Patients interact with the system to view their medical history, medications, and appointments. They may wish to book appointments with doctors, establishing a many-to-many relationship between doctors and appointments. The medical record includes diagnoses, prescriptions, and treatment history, maintaining references to patients and doctors in a one-to-many relationship.

The lab test entity connects the patient, doctor, and lab technician, recording the requested instrument tests, results, and dates. Each user-admin interaction ensures real-time data updates, secure storage, and a smooth hospital flow, enhancing health service provision and improving clinical decision-making.

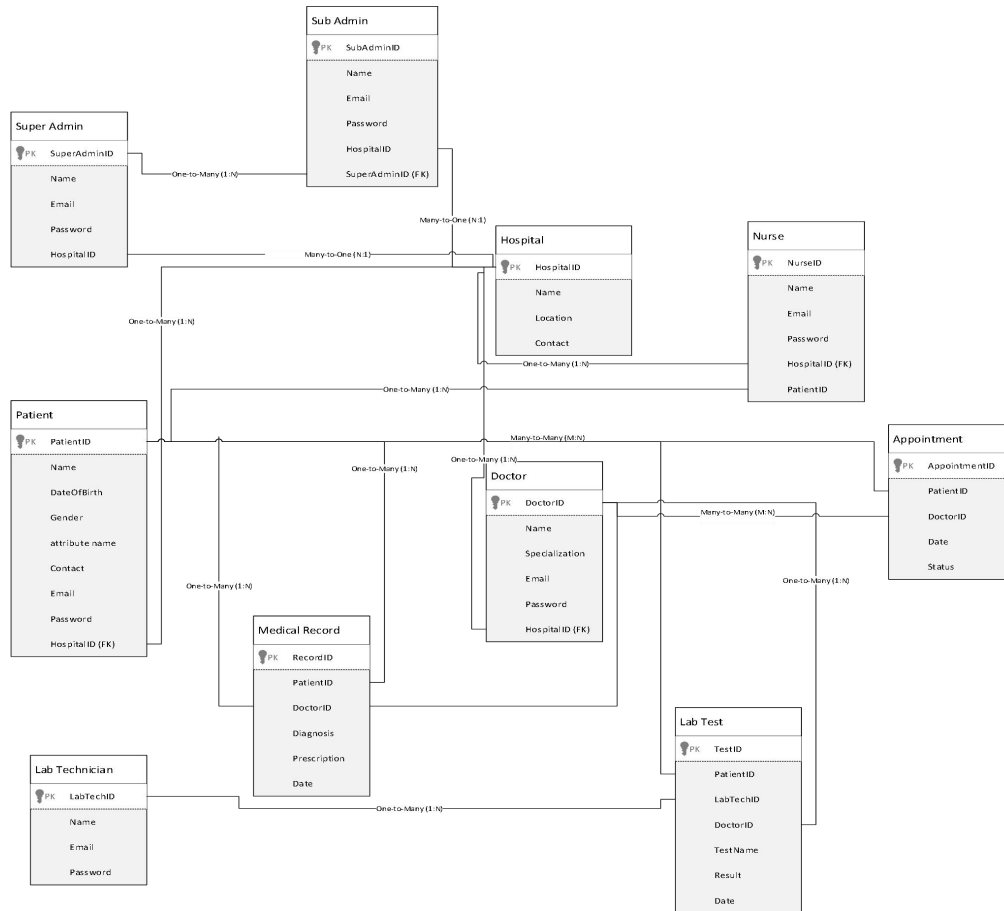


Figure 8: Entity-Relationship Diagram (ERD) of the Proposed System

3.5 Algorithm

The model for predicting hypertension disorder in pregnancy using a machine is given as follows:

3.5.1 Algorithm for User Authentication

Purpose: To validate login credentials and grant access based on user roles.

Steps:

1. Input: username, password.
2. Hash the input password using Microsoft.AspNet.Identity.
3. Query database to match username and hashed password.
4. If match is found:
5. Retrieve the user's role (e.g., Admin, Doctor, Nurse, etc.).
6. Redirect to the corresponding dashboard.
7. Else:
8. Display "Invalid login credentials" message.
9. End

3.5.2 Algorithm for Patient Registration

Purpose: To register a new patient and assign them to a hospital.

Steps:

1. Start
2. Input: Patient name, gender, contact, email, password, hospital ID.
3. Validate input fields.
4. Check if email already exists.
5. Hash password.
6. Insert new record into the Patient table.
7. Output: "Registration Successful".
8. End

3.5.3 Algorithm for Appointment Scheduling

Purpose: To allow patients to request appointments with doctors.

Steps:

1. Start
2. Input: Patient ID, Doctor ID, Preferred Date and Time.
3. Check doctor's availability on that date and time.
4. If available:

5. Create a new appointment record with status = "Pending".
6. Else:
7. Notify user to select a different time.
8. Output: Appointment ID and Status.
9. End

3.6 System Flowchart Diagram

Figure 9 illustrates the System Flow of Centralized Patient Information System. The system's operations are explained from a user authentication stage, accessing the main functionality of the system, and till logging out. The system starts with the Login Page, for user validation. If login information is incorrect, it doesn't save user access and user is asked to enter valid information.

Once the login is successful, the system authenticates the user and opens a specific dashboard according to the user's role (e.g., Admin, Doctor, Nurse). The user

can also navigate to various modules from the dashboard by:

- i Manage Patient Records – for viewing, updating, and storing patient data.
- ii Clinical Decision Support (CDS) – for assisting clinicians in making informed decisions using available patient data and machine learning insights.
- iii Appointment Scheduling – for creating, updating, or canceling appointments between patients and doctors.
- iv Generate Reports – for summarizing patient visits, treatment outcomes, and analytics.

And the user can click back to the dashboard or log out. If the user chooses to log out, the session is destroyed and the system is returned to the login page, terminating the process.

This core cell ensures organized, safe, and user-friendly interfacing with the system and thus enables efficient navigation between modules.

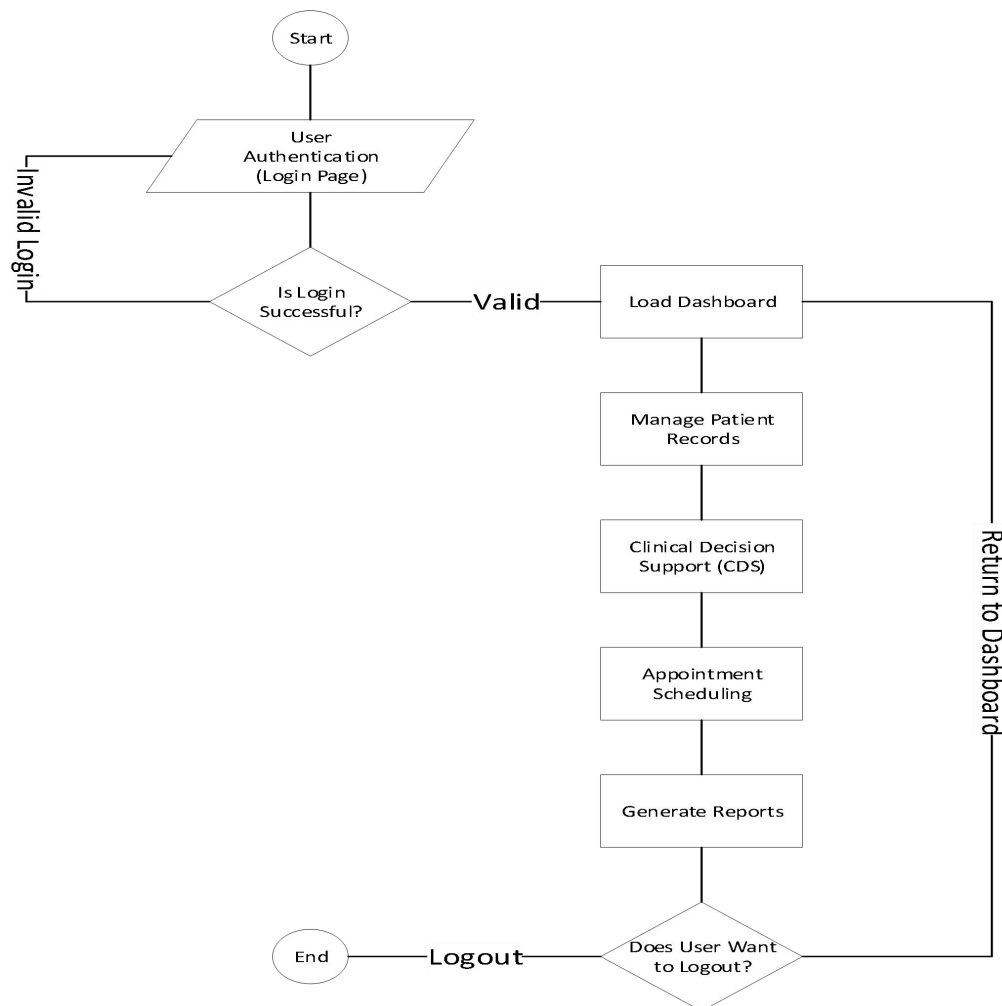


Figure 9: System Flowchart Diagram

3.7 Sequence Diagram for Centralized Patient Information System

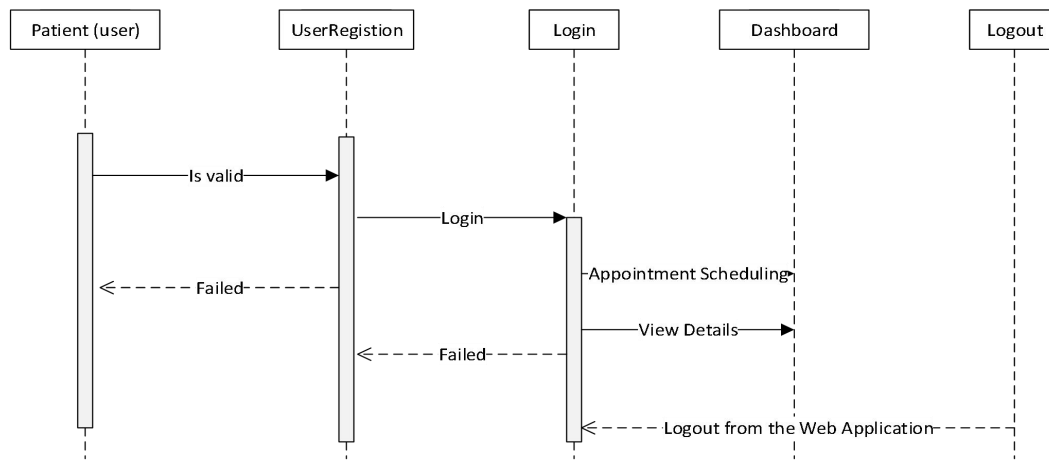


Figure 10: Sequence Diagram for Centralized Patient Information System

A sequence diagram for the Centralized Patient Information System is shown in Figure 10. The user and all other parts of the system are interacting with each other as it is shown in the illustration of the login together with other parts of the system, i.e., dashboard, patient record manager, and database. It illustrates the data being passed from component to component to support authentication, data retrieval, and function execution. It starts from user login and goes up to patient management, appointment booking, and report generation. Each interaction is processed incrementally to maintain cohesion, accuracy, and interactivity of the system.

3.8 System Implementation

The implementation phase involves translating the design of the Centralized Patient Information System into a functional web application.

3.9 Program Development

The Centralized Patient Information System provides a modular and iterative programming methodology to facilitate flexible and hands-on implementation, debugging, and deployment of each component of the system. The system is constructed in ASP. backend logic in C# and frontend interface in HTML, CSS, and JavaScript, ensuring responsive and interactive user experiences. Microsoft Visual Studio is used as the development environment (IDE) for ease. Visual

Studio 2022 was the platform used to script code, build the web application, and debug.

The development is modular and includes modules such as User Authentication, Patient Information Management, Medical Record Management, Scheduling, and Reporting. In this project, each module is developed separately and then combined to make the final system. The solution takes advantage of Microsoft SQL Server to securely store healthcare information and uses stored procedures and parameterized queries to improve both performance and security.

Security functions like password hash, role-based access control, and session management are built at the development stage to guarantee data security and healthcare norms. Unit testing and system testing are performed at different times to verify correct operation, uncover bugs, and ensure that the system meets requirements. Such a systematic way is to avoid building a system that cannot be expanded, is not maintainable, and cannot be changed if enhanced in the future.

3.10 Choice of Programming Environment

The C# language is selected to implement the system (Programming environment). C# is a powerful, strongly typed application development language, perfect for creating secure and efficient applications, such as those used in healthcare data integration.

IV. RESULTS AND DISCUSSIONS

4.1 Desk Check

In the desk check phase, the developer reviewed the software code, line by line, checking for syntax errors, logic errors, and structural flaws. Modules like User Authentication, Patient Information Management, Medical Record Handling, Appointment Scheduling, and Report Generation were reviewed to verify their conformity with coding standards and successful implementation. Documented and resolved any potential issues prior to transitioning to the next phase of testing.

4.2 Unit Testing

Unit tests were also performed to ensure the functionality and correctness of each module. Each module was individually evaluated to guarantee that:

- i. The User Authentication Module correctly validated login credentials.
- ii. The Patient Data Processing Module accurately stored and retrieved patient records.
- iii. The Report Generation Module correctly formatted and displayed patient reports.

Any discrepancies from normal were recorded and adjusted. After a module was successfully tested, it could be assumed stable and integrated with the other components.

4.3 Integration Testing

Integration Testing followed Unit Testing, and was the testing of the Interaction between different modules working together. This round aimed to detect problems in both data flow and communication, and module interoperability. Critical integration areas included:

- i. Interaction between the User Authentication Module and the Database to ensure secure login and session management.
- ii. Communication between the Report Generation Module and other system components to verify that predictions and patient data were accurately reflected in reports.

4.4 Performance Evaluation

The response time (latency) was used to evaluate the system performance for all 10 test cases. Each of the tests consisted in performing common system operations such as login, request of a patient record,

entry of a diagnosis and report generation. This is shown in table 1.

Table 1: Response Times for 10 Test Cases

Test Case	Response Time (ms)
Test 1	120
Test 2	150
Test 3	110
Test 4	130
Test 5	140
Test 6	115
Test 7	125
Test 8	135
Test 9	145
Test 10	105

Average Latency:

$$\text{Average Latency} = \frac{\sum \text{Response Times}}{\text{Number of Test Cases}} = \frac{1275}{10} = 127.5\text{ms}$$

Maximum Response Time:

$$\text{Maximum} = \text{Highest value in the dataset} = 150\text{ms}$$

Minimum Response Time:

$$\text{Minimum} = \text{Lowest value in the dataset} = 105\text{ms}$$

The system's average response time is 127.5 milliseconds, which indicates that it performs efficiently within an acceptable latency threshold for medical information systems.

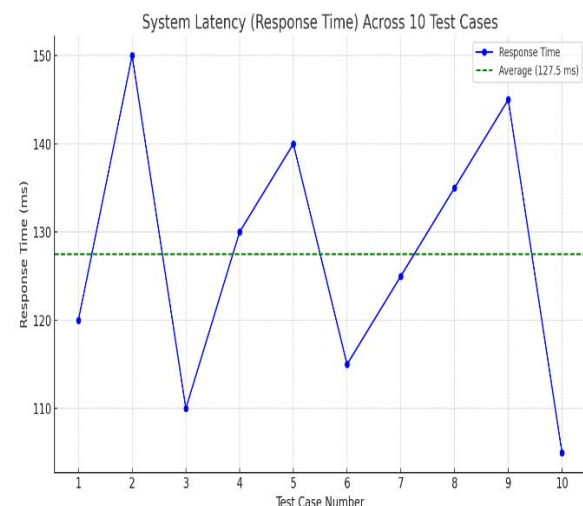


Figure 11: System Effectiveness in Terms of Response Time (Latency)

Figure 11 measured response times varied between 105 and 150 milliseconds with an average latency of 127.5 milliseconds. This value falls squarely within the acceptable bound for real-time health care systems, given that the system's responsiveness impacts medical decisions and the clinical workflow (which itself has some orchestration).

The response times do not experience major variations, indicating the reliability and efficiency under normal operations. The low response time shows that the system can process multiple user interactions with imperceptible delay, which is crucial for critical environments such as hospitals, where the rate of speed at which the data can be supplied directly affects patient care.

Finally, from the latency analysis, the system is very efficient in returning rapid responses, a factor that positively influences its overall usability and user satisfaction. This is consistent with the objectives of the EPA' system to support enhanced health care by ensuring instant, reliable access to patient information.

V. CONCLUSION AND IMPLEMENTATION

5.1 Conclusion

The system was created in ASP. NET C# in Visual Studio for application layer, Microsoft SQL Server for backend data management. It offers role-based access controls, multi-factor authentication, real-time access to patient information, appointment booking, and full electronic health record (EHR) support.

The system was built using object-oriented analysis and design (OOAD) approaches, prototype which was used to collect, analyze how the users liked the system as well as to iterate the process of system construction. Click here to see a list of modules Once developed, modules for different users like doctors, nurses, lab technicians, administrators and patients were created for smooth workflow.

The system's efficacy was tested in a sample of 10 variations of operation time (latency) to prove the point. The test findings also established an average lag of 127.5 milliseconds, proving the capability and agility of the system. In general, the system improves the delivery of health care in minimizing data redundancy, guaranteeing data integrity and empowering clinical decision making.

5.2 Implementation Areas

The Centralized Patient Information System may be used in different clinical settings to optimize workflow and patient care. Major applications include the following:

- i Hospitals and Clinics: For managing electronic health record (EHR), scheduling appointments, recording patient vitals, and sharing care plans across departments.
- ii Government health agencies: Used by Super Admins (e.g., Ministry of Health) for national surveillance, policy planning and resource allocation.
- iii Diagnostic Laboratories: For controlling the test requests, uploading the results and for marriage with the patients' clinical records for timely clinical decisions.
- iv Telemedicine Platforms: Enable remote consults by enabling secure access to patient history and medical information.
- v Pharmacies: For checking prescriptions and verifying medication orders against patient chart.
- vi Academic and Research Institutions: For research, healthcare analysis and training of health professionals.

REFERENCES

- Abell, B., Naicker, S., Rodwell, D., Donovan, T., Tariq, A., Baysari, M., ... & McPhail, S. M. (2023). Identifying barriers and facilitators to successful implementation of computerized clinical decision support systems in hospitals: a NASSS framework-informed scoping review. *Implementation Science*, 18(1), 32.
- Adeghe, E. P., Okolo, C. A., & Ojeyinka, O. T. (2024). Evaluating the impact of blockchain technology in healthcare data management: A review of security, privacy, and patient outcomes. *Open Access Research Journal of Science and Technology*, 10(2), 013-020.
- Ademola, A., George, C., & Mapp, G. (2024). Addressing the Interoperability of Electronic Health Records: The Technical and Semantic Interoperability, Preserving Privacy and Security Framework. *Applied System Innovation*, 7(6), 116.
- Adeniyi, A. O., Arowoogun, J. O., Chidi, R., Okolo, C. A., & Babawarun, O. (2024). The impact of electronic health records on patient care and outcomes: A comprehensive review.

- World Journal of Advanced Research and Reviews, 21(2), 1446-1455.
- Amar, F., April, A., & Abran, A. (2024). Electronic health record and semantic issues using fast healthcare interoperability resources: systematic mapping review. *Journal of medical Internet research*, 26, e45209.
- Ayaad, O., Alloubani, A., ALhajaa, E. A., Farhan, M., Abuseif, S., Al Hroub, A., & Akhu-Zaheya, L. (2019). The role of electronic medical records in improving the quality of healthcare services: A comparative study. *International Journal of Medical Informatics*, 127, 63–67.
- Bala, I., Pindoo, I., Mijwil, M. M., Abotaleb, M., & Yundong, W. (2024). Ensuring security and privacy in Healthcare Systems: a Review Exploring challenges, solutions, Future trends, and the practical applications of Artificial Intelligence. *Jordan Medical Journal*, 58(3).
- Benson, T., & Grieve, G. (2021). Why interoperability is hard. In *Principles of Health Interoperability: FHIR, HL7, and SNOMED CT* (pp. 21–40). Springer.
- Carta Healthcare. (2024, February 20). Centralized, accurate healthcare data enhance the patient experience. [Blog post]. Retrieved from <https://www.carta.healthcare/blog/centralized-accurate-healthcare-data-enhance-the-patient-experience/20/02/2024>
- Cerchione, R., Centobelli, P., Riccio, E., Abbate, S., & Oropallo, E. (2023). Blockchain's coming to hospitals to digitalize healthcare services: Designing a distributed electronic health record ecosystem. *Technovation*, 120, Article 102480.
- Chaparro, J. D., Beus, J. M., Dziorny, A. C., Hagedorn, P. A., Hernandez, S., Kandaswamy, S., ... & Orenstein, E. W. (2022). Clinical decision support stewardship: best practices and techniques to monitor and improve interruptive alerts. *Applied Clinical Informatics*, 13(03), 560-568.
- Lee, T. C., Shah, N. U., Haack, A., & Baxter, S. L. (2020, July). Clinical implementation of predictive models embedded within electronic health record systems: A systematic review. *Informatics*, 7(3), 25.
- Lehne, M., Sass, J., Essenwanger, A., Schepers, J., & Thun, S. (2019). Why digital medicine depends on interoperability. *NPJ Digital Medicine*, 2(1), 79.
- Lewkowicz, D., Wohlbrandt, A., & Boettinger, E. (2020). Economic impact of clinical decision support interventions based on electronic health records. *BMC Health Services Research*, 20(1), 1–12.
- Liao, Z., Wang, J., Shi, Z., Lu, L., & Tabata, H. (2024). Revolutionary potential of ChatGPT in constructing intelligent clinical decision support systems. *Annals of Biomedical Engineering*, 52(2), 125-129.