RESEARCH ARTICLE

DOI: 10.54392/irjmt25324

Asian Research Association

INTERNATIONAL RESEARCH JOURNAL OF MULTIDISCIPLINARY TECHNOVATION

Laravel Technology based Maternal e-Healthcare Systems with Improved Response Time and Stability

Pradeepkumar C Dhage [a,*], Rajesh A. Thakker [b], Krishna K. Warhade [c]

[a] Gujarat Technological University (GTU), Ahmedabad, India.
[b] R & D Cell, GTU, Ahmedabad, India.
[c] Department Electrical & Electronics Engg, Dr. Vishwanath Karad MIT World Peace University, Pune, India.
* Corresponding Author Email: pradeepdhage7@gmail.com

DOI: https://doi.org/10.54392/irjmt25324
Received: 23-03-2025; Revised: 19-05-2025; Accepted: 24-05-2025; Published: 29-05-2025

Check for updates

Abstract: Various maternal e-healthcare systems contribute a great deal to providing holistic care, but most of them are rife with technological constraints that make it impossible for them to address the needs of modern healthcare. The constraints consist mostly of obsolete infrastructures, limited and inefficient integrations, with poor large-scale operational management systems. Literature cited the maternal e-health systems are facing challenges in the area of interoperability, scalability, security, and usability. Key performance indicators to consider encompass response time and stability of the system. Mechanisms in the Laravel Framework are used to address the issues raised by the following: module-based architecture, caching, and queue management. It should however be noted that literature has shown an apparent gap of research that has greatly impacted on response time and stability in maternal e-health systems. Additionally, there is a critical lack of slim systems broad integration that's driven by data. To fill these gaps, this research is intended to strengthen the two performance indicators by means of Laravel use. The research enhances the given e-healthcare system, to increase performance outcomes regarding response time and system reliability, through the use of the Laravel framework, on the modules including user authentication, data collection, and report generation. Detailed performance measures to be studied will include load, stress, and endurance testing with JMeter in response time, and throughput, and error rate settings. The assessment of system performance based on the functionality of Laravel focused on the following metrics concerning the predetermined service-level agreement: mean response time, throughput, and error rates. The results from the analyses are merged with the graphs and tables which indicate an improvement in performance.

Keywords: Performance Indicator, Response Time, Maternal Ehealth System, Testing, Jmeter Tool, Load Test, System Performance

1. Introduction

E-healthcare systems are fundamentally important in providing patients info with complete and efficient care. They lead the way to computerized management and maintenance of electronic health records and represent potential for improving clinical decisions for the healthcare professional, as they promise to improve efficiency, reduce medical errors, and improve patient outcomes [1]. However, the successful implementation of these systems remains a significant challenge, particularly in resource-constrained settings [2], including obsolete IT infrastructure, a lack of integration, and inefficiencies in managing large-scale operations [3]. These challenges become even more significant if the healthcare system is designed for maternal care. The maternal care e-health system needs high security, steady performance and optimum stability in response to effective patient management [4]. It has a direct impact on maternal health data, and other stakeholders involved i.e. hospitals and pathology labs that shows the need of performance analysis. The capacity to react quickly and precisely resolve user concerns, which have been directly connected to the best possible patient care delivery is a crucial component of e-health systems performance [5, 6]. The identification of response time and throughput as essential component is crucial for understanding productivity and use reliance [7].

The maternal e-healthcare framework MAMATA CPS, which is based on Laravel technology and includes enhancements in security, scalability, and data management capabilities, was presented in earlier studies [8]. The research work addressed important issues including validation, reliability of data by proving strong security aspects. For effective real-world deployment and use, however, it is also crucial to provide system stability.

With a focus on key measures such as response time (RT), stability (error rate), throughput, the author's research work focuses on the performance evaluation [9] of proposed system. Considering these targeted measures, authors attempt to do load and stress testing by modelling numerous users in real time simulation to study system behavior in different demand circumstances. Eventually the study measures to compare with well-known performance practice standard i.e. Service Level Agreements (SLA) [10] to understand the depth of stability and improvement (if any) required. This research seeks to comprehensively acquire key indicator matrix of MAMATA CPS in strategic manner to enhance maternal e-healthcare systems stability. Furthermore, the findings will provide information for creating stronger digital health infrastructures for better outcomes in maternal health.

The paper discussion is organized as follows. The literature review of performance metrics in healthcare systems in 'Section 2'. In 'Section 3', Methodology pertaining to MAMATA CPS is discussed with Laravel as framework, technical architecture and databased design. The testing approach and related test setup has been presented in 'Section 4'. The results and findings are provided in 'Section 5'. Finally, the result analysis and related discussion is reported in 'Section 6', followed by conclusive remarks are provided in 'Section 7.

2. Literature Review

Advancements in digital technologies have shaped the trajectory of maternal e-healthcare systems; however, there are still considerable gaps in their design, integration, and performance. A number of scholars have investigated the challenges and opportunities facing maternal healthcare systems, creating a foundation for the development of new frameworks (e.g., MAMATA Cyber-Physical System). This section reviews the literature pertaining to maternal e-health systems, including limitations and performance measures, possibilities for Laravel technologies, and the design considerations necessary for success.

2.1 Current e-healthcare Systems and Challenges

The collection, administration, and analysis of maternal and child health data depend heavily on maternal e-health systems. On the other hand, current platforms frequently depend on antiquated technology, which renders them ineffective and vulnerable to integration issues. For example, hospitals, clinics, and pathology labs are among the various stakeholders whose capacity to transmit data seamlessly is restricted by the absence of interoperability among systems. Standardized data formats and real-time monitoring are essential for facilitating prompt clinical choices and improving mother and child health outcomes, according to studies.

Critical analysis comprising the strength and weakness/scope for several web based medical care systems has been summarized in Table 1. The Sivan and Zukerman system discuss two additional important concerns in e-health systems i.e. protection of one's privacy and security [11]. Due to the complex and variety of data formats of health information, these systems are vulnerable to data breaches and unauthorized access unless severe safety precautions are not in place. Furthermore, study of 'Ayu & Nuryasin' highlighted that low adoption rates are frequently occuering due to sloppy user interfaces, especially in short of resources and remote regions where the need for simple and accessible solutions is critical [12].

An open-source medical record system for developing nation named 'OpenMRS' (an Java-based Platform) has been described in [13]. This platform is modular and has a large data handling capacity, but its adoption requires complicated infrastructure. An advance study shown that Hospitals with limited resources can use an integrated EMR system (Bahmni) [14] that makes use of Angular technology (front end) and OpenMRS, Odoo (back end). Having a user-friendly online interface and strong connectivity with other systems, this system attempted to integrate clinical records and hospital administration. But when interoperability is considered, this system becomes more complicated with a multi-component architecture that necessitates regular upgrades and localization.

System developed on a commercial telehealth app called Babylon Health [15] offers, AI-based medical consultations and symptom checking. It uses Node.js to provide real-time capabilities and approaches AI integration for diagnostics, which calls for highly scalable cloud-based infrastructure. This raises issues with data privacy and regulatory compliance, as well as the high setup and maintenance costs. A requirement of health data dashboard is important at-least for small scale clinics to handle the medical data, which is reasonably attempted by system named 'Django' i.e. a typical health information system [16]. Considering python-based system, it may provide rapid development with built in admin tool platform, with high security, however this requires third party packages for healthcare -specific needs and shows less flexibility.

All above major studies/system found to be working in medical target areas. Moreover, it is noted that these systems, applications target general patient health record (i.e. EHR). Whereas, in this paper, we have targeted customized healthcare system limited to maternal women record (pre-natal).

Table 1. Critical synthesis of web based medical systems available in healthcare ecosystem

System / Study	Short Description	Strengths	Weaknesses / Challenges or Future Scope
Sivan & Zukarnain (2021) [11]	Brought up security and privacy concern of cloud-based health systems.	Makes people more aware of important legal requirements and possible threats	Not designed for 'Maternal ehealth records'. Database, data safety and standard security frameworks for all systems are called for.
Ayu & Nuryasin (2021) [12]	Author discussed how inadequate (poor) UI design impact at rural-remote areas.	Articulately focuses on practical functional-usage challenges and well pointed user-concentrated design demands.	Accessibility needs improvement especially in rural regions.
OpenMRS (Java-based platform) [13]	Strong data handling, open-source medical record system for developing countries. Modular design.	Scalable in nature, helps to facilitate custom adaption, large data capacity.	May Explore design for specific to 'Maternal ehealth system', and database. Demands complex deployment of setup.
Bahmni System. (developed on OpenMRS, Odoo and Angular) [14]	Clinical and hospital administration data are combined in an integrated EMR. Built on Angular frontend, with use of OpenMRS, and Odoo platform at backend	Easy-to-use UI, robust integration across departments.	Not designed for 'Maternal ehealth system' and database. Complex multi-component architecture, needs frequent updates.
Babylon Health [15]	Uses Node.js. AI-powered telehealth system adopted for real-time consultations and symptom checking.	Real-time services, strong AI capabilities, modern web stack (Node.js).	Expensive, Privacy / regulatory concerns. Demands complex deployment setup
Django-based HIS (Python-based) [16]	Easy to custom health information system using Django; makes it suitable for smaller clinics.	Exclusively designed admin tools, quick development, secure.	Available for General patient care, not designed form specific maternal database structure, needs third-party healthcare packages.

This is continuous nine (09) month prolong data handling, where various tests and data formats introduces in each trimester. The Maternal health record formats need lots of standardization and have complex parameters.

2.2 Key Performance Metrics in Healthcare Systems

The current research emphasizes various essential performance measures for healthcare information systems, such as response time, system uptime, and user satisfaction [17]. For application system related to mobile health (m-health), which need to obtain data to make choices with instant response, needs truly instant reaction time [18] this makes even more crucial in healthcare ecosystem. Another study literature emphasizes the importance of adaptation and flexibility as key factors influencing the healthcare systems [19].

Considering above discussion, a swift response time becomes de facto requirement to achieve system stability. An unstable system can diminish productivity, postpone patient treatment, and disrupt clinical operations which is not acceptable in healthcare ecosystem. Hence healthcare systems must be stable to deliver continuous access to critical medical information and services [20]. Timely responses are essential for guaranteeing immediate access to information for medical professionals and patients [21]. A sluggish response affects significant patient care that impact on overall e-healthcare system. This gradually points the serious role of response time and throughput, error rate.

The Authors expressed concerns on system service level and time to react in [22]. Additionally, inadvertent breakdowns of the system, with exceeded durations, were observed in separate work citing to utilization of e health records for just-in-time preparation aimed at ensuring healthcare services smooth execution [23].

In health care system, it is also important to describe how the system handle data which moves between location inside its block. This bring out one more important indicator i.e. 'Latency. For effective workflow, secure patient care, and real-time decision-making, minimal latency is essential [24]. Reduced users' satisfaction, misdiagnosis, and treatment delays probably came from high latency. However, the distance between data sources and processing units (crucial in telehealth care contexts), server load, database request intricacy, network outages, poor scripting, are some of the variable influencing latency. For instance, a delay of 18 to 39 ms was identified in study on real-time IoT-based patient monitoring systems [25].

Maintaining continuous access to critical medical data and services require a high system uptime (i.e. the proportion of time a system has been functioning and usable). It is a gauge of the stability. Downtime can interfere with clinical operations, delay patient treatment, and disturb workflow [26]. Poor hardware setup and weak code may invite cyber attackers that eventually affect system up time. As per one research review in [27], within the first year of deployment, an infrastructure was up and running more than 99 percent of the time. To mitigate system stability, maintaining and updating the system on a regular basis along, employing strong security protocols to stop cyberattacks, and creating thorough downtime protocols has been suggested in [28].

An assessment-based review emphasized how crucial it is to create operational backup plans in order to maintain care continuity in the event of system failures. A thorough examination of system outages in ehealth systems addressing the base reasons of system failures, encompassing both organizational and individual facets [29]. A performance measures often evaluated through performance test to understand system's behaviour under various situation and user load is discussed in [30].

The aforementioned research shed light on the critical significance of response time and system stability with regard to the ehealth system considering the level of satisfaction experienced by healthcare service provider and the end users.

2.3 Frameworks in e-healthcare Systems

For healthcare systems to be effective, quick, and secure, the frameworks are crucial resources. They streamline the development process by offering pre-written script and saves developers time [31]. It encourages scalability, which facilitates comprehension, modification, and long-term maintenance of the codebase.

The available frameworks are Laravel, Node.js, Django, and Java Spring. Laravel, a PHP framework have resilient functions with simple, ease to ready syntax. It offers a strong option for the development of healthcare applications. The Model-View-Controller architectural pattern adopted by Laravel facilitates an individual division of concerns, resulting in codebases that are more maintainable and scalable. This is particularly important in the complex field of healthcare software, where modifications and enhancements occur frequently [32]. Laravel incorporates object-oriented programming principles, enhancing the scalability of the application [33]. Selecting a backend infrastructure for healthcare applications necessitates thorough evaluation of factors including security, scalability, maintainability, and the accessibility of qualified developers.

Table 2 depicts the broad level comparative among the different frameworks. Laravel is compared to its alternatives in order to determine its suitability for e health system. Node.js is an excellent choice for real-time applications. This makes it an ideal choice for telehealth platforms or other systems that require instant changes [34]. Django is a Python framework that provides possibilities for rapid development as well as a variety of built-in functionality. These features include an object relationship manager (ORM) and an administrative interface, both of which can speed up the process of developing data-driven healthcare applications.

Table 2. Framework comparison

Framework	Pointers to consider	Suitability
Laravel	Python based, Elegant syntax, Modular and secure, MVC structure, Object oriented programming principles. Scalable in nature	Ehealth system (Large, medium, small)
Node.js	Java script, Non-blocking architecture, Event-driven	Real time application, Microservices, Telehealth platforms
Django	Python based, Rapid development, Object relationship manager.	Healthcare application (data-driven)
JavaSpring	Java based, Enterprise grade framework, Heavy to deploy, Complex in infrastructure.	Only for large scale health application,

Java Spring is a mature and enterprise-grade framework that offers exceptional scalability and performance. As a result, it is ideal for large healthcare organizations that have demanding workloads. However, due to its complexity and steeper learning curve, it may present difficulties for smaller teams or projects that have limited resources.

3. Methodology and Framework Implementation

This section focuses on the implementation strategy adopted, considering the system architecture, cloud infrastructure and setup, how the database is designed, structured for smooth handing of the complex maternal women data and last user interface development. The Laravel framework serves as the foundation for the MAMATA Cyber-Physical System (CPS), which is an innovative technological solution, wherein its execution strategies designed to address key issues with data privacy, efficiency, and performance with quick response. This section highlights the ability of Laravel technology to address performance issues by discussing the framework's technical architecture, database design, performance measurements, and implementation techniques.

Response time and throughput were among the performance metrics evaluated for conditions, and the results showed that the system can handle a variety of user accounts without experiencing excessive response times. The MAMATA CPS framework establishes a measurable solution for maternal healthcare needs and experience in e-health systems.

3.1. Laravel as a Solution Framework

Laravel, an open-source PHP framework, has gained recognition given its strong security features, modular design, and scalability. The development of maternal e-health systems has advanced significantly with its incorporation into the MAMATA CPS framework [35]. Laravel facilitates swift to integrate various data sources and, enabling seamless interoperability. Further, the privacy issues common in maternal e-health systems are addressed by its integrated security features, which include encryption and role-based access control.

In addition, Laravel framework prioritizes user experience, offering customized user interface (named LaraAdmin) designed to meet the needs of various stakeholders, including maternal women, healthcare providers, and administrators. This resonates with literature on the need for user-friendly and intuitive system design to facilitate user uptake and use. Figure. 1 depicts the idea of MAMATA CPS model, underlining its role in data integration, ensuring secure and efficient data management. As per Figure 1, the prime hub connecting all the stakeholders is the MAMATA CPS system.

This system is aimed to perform, with its crucial role as- 1. Data integration: Bringing together data from multiple sources on one platform.2. Information Exchange: Facilitating the safe and easy exchange of information between various stakeholders. 3. Coordinated Care: Encouraging improved treatment coordination by providing comprehensive patient data understanding all relevant stakeholders.

3.1.1 Framework Architecture (NMDBS Technical setup)

Given its sturdy and modular design, the MAMATA CPS framework makes use of Laravel technology followed by MVC structure. The technological architecture of the NMDBS server system is depicted in Figure 2, highlighting the role of diverse components.

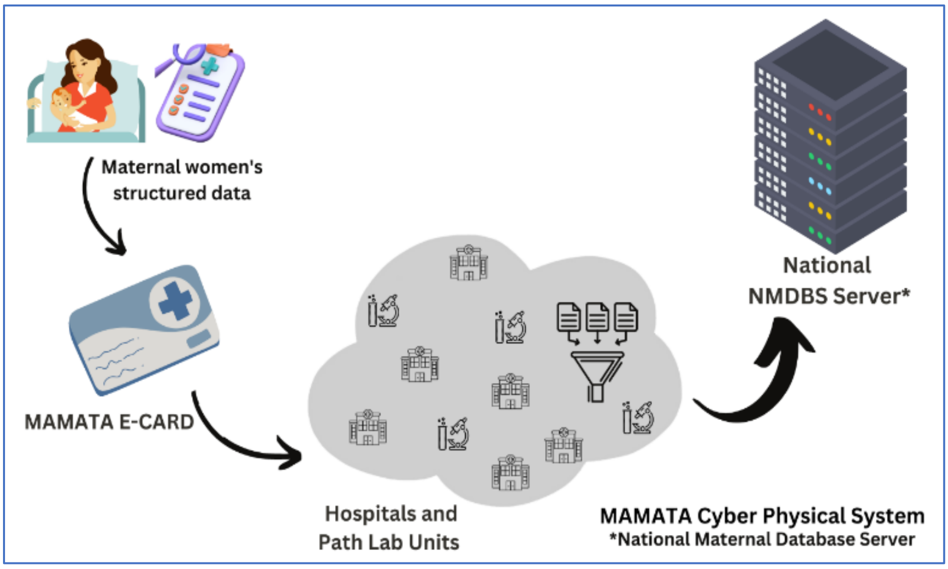

Figure 1. MAMATA CPS Conceptual model for Maternal E-Healthcare Systems

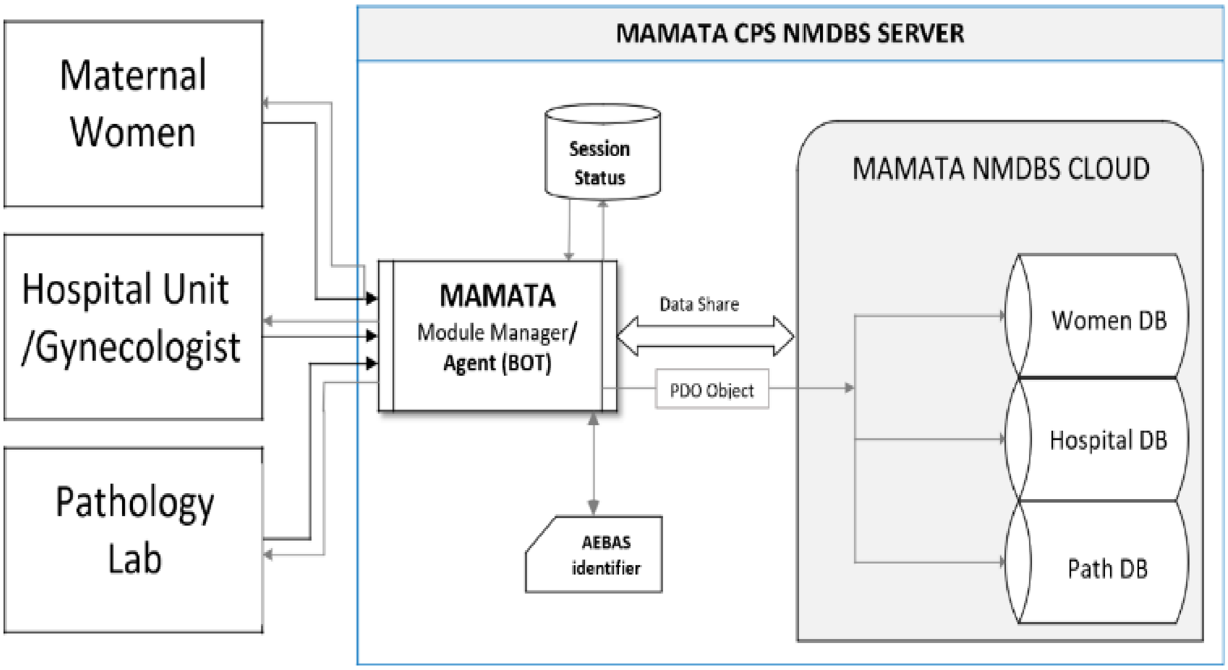

Figure 2. MAMATA CPS System Architecture.

- *National Maternal Database Server (NMDBS):* Centralized repository for all maternal health data. NMDBS implements secure storage protocol guidelines i.e. OWASP (Open Web Application Security Project) [36], ensuring compliance with data privacy regulations.
- *MAMATA Module Manager/BOT:* It acts like controller unit module that intermediate between data sources and the National Maternal Database Server (NMDBS). Its responsibilities include data acquisition, validation, and secure transmission adhering to a list of typical security flaws that might impact online applications i.e. known as the OWASP Top 10 vulnerabilities. Referring to the authors security associated work [8, 35] the security methods developed using these vulnerabilities to safeguard apps, while ensuring data standardization across diverse formats and secure integration with NMDBS. This controller responsibilities are critical in ensuring the seamless integration and secure management of data across the maternal ehealth ecosystem.

  This controller is been made available in path snippet-1 below.

Snippet-1

app/Http/Controllers/Controller.php

Snippet -2

The snippet of the controller is given in 'Annexure' (as code snippet -2)

The controller (i.e., MAMATA Module Manager) manages all activities related to user request processing, including data collecting, request validation, and secure data transfer. The data collecting process collects information from expectant mothers, healthcare facilities (electronic health records), and pathology laboratories (laboratory reports). It presumably does essential functions such as data validation to guarantee data integrity, and security measures to safeguard sensitive patient information. Subsequently accountable for converting data from various forms into a standardized format compatible with the NMBDS, while ensuring reliable and secure transfer of data to the NMBDS. Hence it prevent data loss and unwanted access.

- *Middleware Management:* The logic to process and reply to incoming requests is contained in controllers, which manage the application's HTTP requests in Laravel. However, in contrast, middleware filters these requests by intercepting them before they reach the controller. This allows the middleware to carry out tasks like authentication, validation, or request modification. As illustrated in Figure 2, a PDO (PhP Data Object) is used to run SQL queries, insert data, retrieve results, and carry out other database operations. The middleware functionalities are taking care of encryption, redirect authentication and csrf token verification.

This path of middleware handle is defined at below snippet 3 with file directory -

Snippet-3

app/Http/Middleware/EncreptCookies.php

app/Http/Middleware/RedirectAuthentication.php *app/Http/Middleware/VerifyCsrfToken.php*

The snippet of the above all three function of middleware management are collated in ‘Annexure’ section (as code snippet-4).

The snippet-4, represent major classes for encryption, redirection and csrf token verification. Encryption class takes care of exclusive names of the cookies that should not be encrypted. Incoming request handling using string ‘guard check; to redirect and allow authentication is being taken care by class ‘RedirectIfAuthenticated’ and class ‘VerifyCsrfToken’ checks the URI (unified resource identifier) that should be excluded from csrf verification.

3.2. Cloud Infrastructure Setup

The framework's primary part is the National Maternal Database Server (NMDBS), designed for scalability and security, employs encryption and role-based access control to safeguard sensitive data.

Its infrastructure designed with average scale testing in mind that include major hardware with-

- Cloud service: Digital Ocean
- Storage: 25 GB Disk
- Memory: 1TB of Memory in use.
- Number of CPU: 1 CPU under use at cloud
- Operating system: Ubuntu Host deployed over cloud
- Additionally, Frontend UI (named LaraAdmin) powered user interface ensures usability, providing healthcare professionals and patients with responsive and mobile-friendly access. It doesn’t rely on APIs such as OAuth2 or REST API. LaraAdmin UI Framework: is designed by author and incorporated in system that has customized UI for administrative tasks which facilitates role-based access control, CRUD operations, and module management [37].

3.3 Framework Design and Core Components/Modules

The MAMATA CPS framework is structured around several key components to ensure performance, scalability, security, and user-friendliness. These components are designed to address the specific requirements of maternal healthcare systems, Figure. 3 shows the core components required to implement e-health system. The MAMATA CPS is designed on the same. These core components are described below,

- *Patient Governance Component:* This component manages patient enrollment, demographic information, clinical history and provides healthcare professionals with a simple way of accessing and organizing patient information.
- *Data Integration:* Facilitates integration of information from key contributors (e.g., pregnant mothers, hospitals, and pathology labs), including electronic health records, and clinical documentation; into a centralized, standardized format for ready access and evaluation.
- *Maternity Care Module:* Enables antenatal care, scheduling of appointments, and monitoring parameters for the wellness of mothers, for example, evidence-based clinical guidelines to aid healthcare professionals with informed decisions.

Data Analytics and Reporting: This module enables top-level analytics to determine trends, monitor outcomes, and inform policy-making in maternal health.

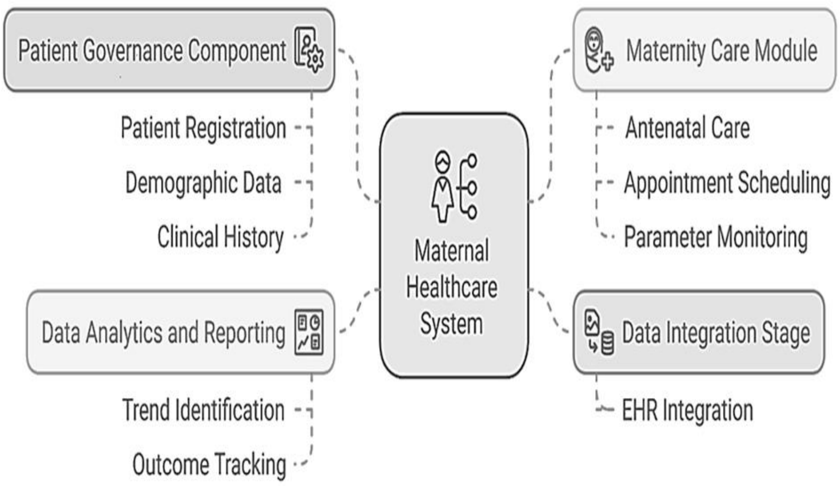

Figure 3. Core components for Maternal E-health system

3.4 Database Design and Model

To manage diverse, unstructured maternal and child health data, the proposed system utilizes a MongoDB SQL database; addressing standardization and interoperability challenges in maternal healthcare records, thereby enhancing the overall system's functionality. The proposed structure is based on the "MAMATA MATERNAL RECORD TEMPLATE" [38], advocates for a unified and standardized approach to data collection and organization.

Database normalization is essential for organizing data to reduce redundancy and improve data integrity. The database is normalized to the first and second normal forms. First Normal Form (1NF) guarantees atomicity by eliminating redundant groups. Partial dependencies are eradicated by the Second Normal Form (2NF). NMDBS also focuses on migration management. This is been taken care at path, snippet-5.

Snippet-5

mamata/database/migrations.

While the database seeding is done at path, as shown in snippet-6

mamata/database/seeds/databaseSeeder.php .

The seeding is important to test the application that helps developer to have quick understanding of database behaviour response. Snippet-7 (mentioned in annexure) shows use flow for database seeding at NMDBS.

Refereeing to seeder code (i.e. snippet-7 of annexure) several models are constructed to optimize database handling for quality, simulation, security, and scalability. Frontend UI (LaraAdmin) adopts simple and easy to access dashboard module in seeder code. The following database design factors are discussed below,

a) The goal of data quality is to maintain accuracy, consistency, and completeness by appropriate limitations, rules, and referential integrity checks.

b) Data simulation involves creating models and schemas to represent patient profiles, clinical notes, test findings, and other relevant data of mother and child health.

c) To protect sensitive patient data, stringent security mechanisms such as audit trails, encryption, and access limits are implemented in compliance with data privacy legislation.

d) Scalability: The database must maintain accuracy and performance while handling growing volumes and complexity of pregnancy and child health data.

Moreover, the database structure advocates for a unified and standardized approach to data collection and organization. Categorizing key elements including:

3.4.1 Standardized Data Formats

Data fields such as patient demographics, medical history, and clinical parameters adhere to internationally recognized standards, including ICD codes [39] for diagnoses.

I. *Patient Demographics & Identification:* Patient Information, Spouse information.

II. *Medical & Pregnancy History:* This section of the maternal record template necessitates

Standardization for data consistency, interoperability, and substantive analysis, requiring a Common approach to data collection and documentation.

- *Maternity History Model*

The class uses standardized terminology and coding systems like the WHO ICD to store data on past pregnancies, deliveries, and complications, ensuring consistency and facilitating data aggregation within healthcare facilities.

The Maternity history model is made at path (snippet–8)

Snippet-8

app/Models/MaternityHistory.php .

Important parameters are represented in a class diagram strategy, as shown in Fig 4. with each parameter (i.e. +string) being a function of the class.

- Gravidity and Parity: The gravidity (total pregnancy times of women) and parity (no of pregnancy delivered past 20 weeks of viable gestational age) are recorded using the standardized G_P_ system [40] and included as attributes in the MaternityHistory class.

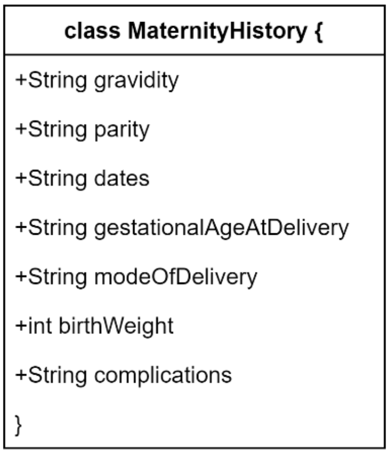

Figure 4. Standardization of records class diagram (Maternity History).

The ‘MaternityHistory’ class also records dates with consistently specific format (e.g. YYYY-MM-DD), gestational age at delivery (in weeks and days), mode of delivery (‘Natural’, or ‘Cesarean section’), birth weight (in grams), and complications (if any) using standardized medical terminologies and classifications like ICD codes, with the delivery mode using standardized terms like "Normal-Natural Delivery."

- Class ‘MaternityHistory’ covers the major parameters as discussed above and the snippet of the same is given in annexure, i.e. snippet-9 that covers all key parameters referring to the ‘mothers id’ only.

[iii] Clinical Data & Management: The maternal record ‘Clinical Data and Management’ segment consolidates significant information regarding the present pregnancy and associated medical interventions.

Routine data in this section is significant in comprehending the patient's health status, care plan, and treatment trajectory. A uniform and systematic process of data recording in this segment is significant in ensuring consistency, facilitating interoperability, and enhancing the credibility of data analysis. The major key categories under clinical data and management is elaborated with its subsequent parameters considering class diagrammatic approach as shown in Figure 5.

Test and Investigations The separate model that cover test related medical data of maternal women. To realize it a separate ‘Test class’ has made to store standardized information about medical tests, including test names, results, and dates, enabling consistent recording and analysis using LOINC codes [41]. The test model (i.e. class) has been setup at path –

Snippet-10

app/Models/Test.php.

Wherein this model (a special class for testing) takes care of many tests involve in maternal care. A code snippet-11 of the same is given in annexure.

Key standardized parameters including Test Names using standardized terminology and widely accepted medical terminologies for tests like FBE, Hept-B, HIV, GCT/GTT (Glucose Challenge Test/Glucose Tolerance Test), and GBS.

The ‘TrimesterCheckup class’ captures information related to each checkup in the trimesters, such as measurements such as fundal height, fetal heart rate, and blood pressure, in controlled vocabulary and standardized units as shown in Figure. 5. Complication data that is encountered during pregnancy is captured using a ‘Complication class’ using ICD codes, and Medication and Intervention data is stored in the 'Doctors' remarks', which are stored as string values. Data storage is made secure by encrypting, auditing, and using access controls, and the system is scalable to support increasing amounts of data without performance degradation.

3.5 Server Module Roles and Permissions

The MAMATA CPS framework encompasses several key modules and features to address the diverse needs of maternal ehealth. National Maternal Database Server (i.e. NMDBS) aims to update maternal treatment care information for each maternal female's account by the authorized Hospital /Pathology lab personnel (could be a doctor or another valid representative).

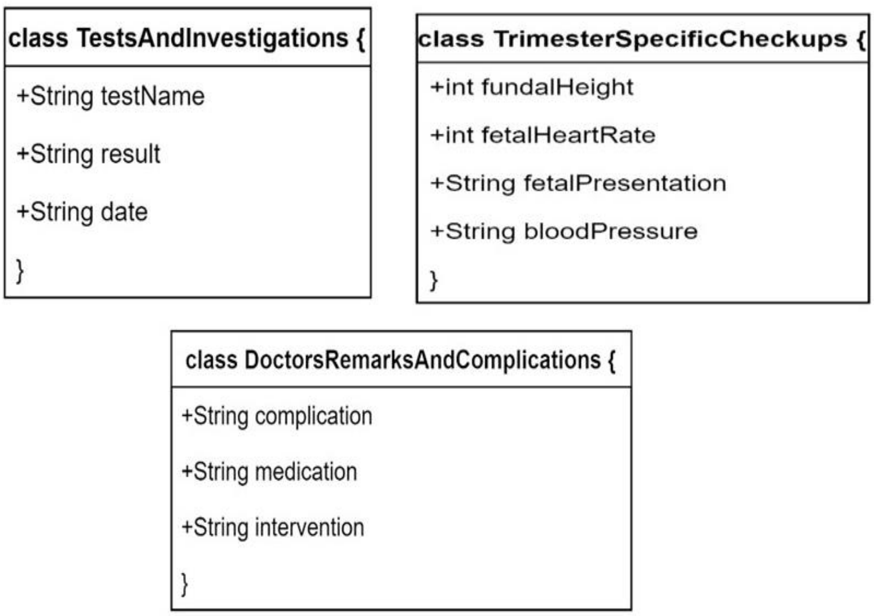

Figure 5. Standardization of records class diagram (Clinical data and Management).

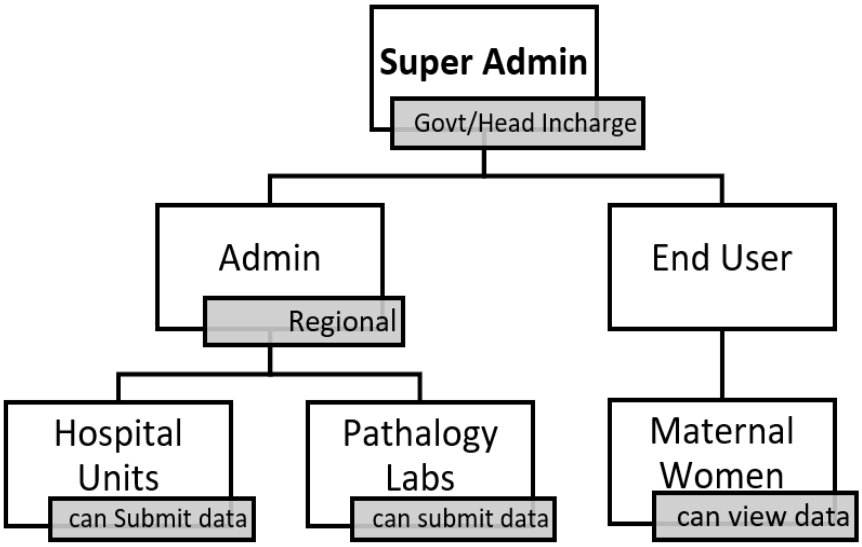

Figure 6. Server Module Roles and workflow

MAMATA NMBDS Server User's roles are divided into Super Admin, Admin, and End User categories. The users are defined in user.php of Laravel as shown in path 1 below. Wherein the roles of the same is depicted as in code snippet 12, as shown below.

Snippet-12

$$app/User.php$$

Each role specifies unique permissions for accessing account data. This server enables every user to create an account by using a specific national registration number (such as UIDAI/Aadhaar No in the case of India). The individual in charge of "Admin" duties (known as the authority for regional hospital/pathology lab) can modify and send the maternal woman's care details following the specified format on the MAMATA NMDBS server, but only after approval by the super admin (an official appointed by the government of the respective country). The snippet 12.1 (reported in annexure) highlights how the role indexation made.

In cases where maternal women hold End User rights, they are only permitted to view their maternal treatment information on their account by providing valid login credentials. The complete process for roles and permission access is demonstrated in the Figure 6.

The NMDBS provides a strong role-based access control system. The MAMATA CPS system is fully managed by the Super Admin Module. Super Admins can do several system tasks viz. User and Role Management: Creation, updating, and deletion of user accounts and role and permission assignment. Managing users and roles entails creating, editing, and deleting accounts and assigning roles and permissions. MAMATA CPS module configuration requires setting up and customizing modules and functions. System Settings: Manages system preferences and setups. Additionally, observing and updating system logs, analyzing data, and creating detailed reports.

Admin module: Helps healthcare administrators and managers manage their duties. manage pregnant women's healthcare information, including registration, prenatal treatment, high-risk pregnancies, postpartum care, and vaccines. It collects and analyses maternal health data from various sources, tracks key variables, and draws conclusions. Users receive access privileges and permissions based on their responsibilities.

The User module serves pregnant women, healthcare providers, and support staff as system users. It lets users examine and manage their personal health information, including medical information, test results, and treatment plans. Users can also arrange doctor appointments and use virtual health services. They receive customized health advice based on their health profiles and needs.

4. Test Setup (limited to response time and stability)

To implement the framework while considering the objective of performance matrix, a strategic evaluation is required. A major performance key matrix like ‘Response time’ (i.e. RT- i.e. how fast a system responds to the user’s query), and error rate (how well a system executes all query without any drop). An appropriate test setup configuration could be used to do both, which demonstrate system stability. As part of the test setup, a number of simulated user sessions are run, and the behavior of the system under various loads is observed. To create realistic user traffic and get pertinent performance data, load and stress testing tools like JMeter or Siege used. However, before best fit test execution, a comprehensive system implementation strategy also needs to look over i.e. use of technology stack, system behaviour control, and User Interface optimization etc. These key steps are discussed below:

- Technology stack- Authors chose to leverage 'Laravel's modular architecture' and integrate reliable routing and middleware solutions for safe and effective data processing.
- Role based access control- Implementing role-based access control using the Zizaco/Entrust packages helped to manage access permissions effectively. Entrust happens to be flexible option to add 'Role-Based Permissions' [42].
- Optimized User Interface (UI) development: The project focuses on developing an optimized UI, utilizing LaraAdmin built-in caching mechanisms such as 'Redis integration' [43], to enhance usability and streamline administrative tasks, while reducing database query loads.

4.1 Load/Stress Test Setup-

To establish a baseline for the system's performance, a comprehensive evaluation is being conducted to monitor critical metrics, such as response time, throughput, and error rates. Several user session simulations have been performed as a part of the process, and the system's response to various loads has been monitored.

To generate meaningful user traffic and get crucial data on performance, load and stress testing tools like JMeter [29] has been used. The efficacy of the MAMATA CPS framework is evaluated utilizing key parameters, including: a) Users Count: The quantity of virtual users utilized for testing. b) Count of Transaction: alternatively referred as no performed activity. c) Response Time: Reflects the system's efficiency in executing user transactions. d) Throughput: Determine whether or not the system is able to efficiently manage the needs of multiple users at the same time and e) Error Rates: Monitors deviations from the norm in order to guarantee dependability and conformity with customer service level agreements (SLAs).

When it comes to the technique of the MAMATA CPS framework, integration, performance, and security are given absolute attention. The comprehensive technical architecture and systematic implementation plan provide a patient-centered and data-driven approach to maternity care. The proposed framework addresses the shortcomings of traditional maternal e-health systems by offering scalability, robust security, and seamless data exchange. Moreover, the test environment's outcome and related observations are shown in the next section i.e. 'Test Results'.

5. Test Results

The methodology discussed above has high attention towards performance analysis of the suggested framework, the essential parameters for consideration are average response time, throughput, error rates, and compliance with service-level agreement (SLA). These metrics are assessed through load and stress tests performed over the proposed framework. This section discusses the results of these tests, supported by graphical representations and tabular data.

5.1 Load Test Results

The load test was conducted using JMeter tool [44] with varying user loads (200 and 300 virtual users) to evaluate the system's response under typical and high-demand scenarios. When examining the response, the following major transactions/events are considered: 1. Launch site (once user sent request to server), 2. Create account (user request initiated to create account) 3. Add details (user request for details submission) and 4. Logout (user request to sign-off). The behaviour of system against these major transactions have been observed and captured with key metrics include average response time, throughput, and error rates, and '90% Line' [45]. Some key test features are discussed as below-

a. Avg Response Time: It is time to response for every sampler/transaction request. For Samples with the specific label, it is the math average of all response times.

b. 90% Line: The term "90 line" denotes the 90th percentile number, which indicates that 90% of the observed response times throughout the test came below this particular time, meanwhile only 10% of the responses exceeded it.

c. Service Line Agreement (SLA): It is and understanding agreement between service provider and end user. It is about setting expectations for a performance test and the standards the developer/supplier needs to comply with. Generally, it is decided mutually between developer and client/user. In an SLA commonly, consideration system response should not be more than 3 to 4 seconds [46].

5.1.1 Test at 200 Virtual Users

As per the test setup, the authors have applied simulated load of 200 concurrent virtual users (VU). The initial measurements of the system's performance revealed that the average response time for key transactions was significantly above the established Service Level Agreement threshold (setup as max 2000ms by author). The system performance has been observed against major four (04) transactions/event i.e. Launch site, Account creation, Details submission and Logout event (Table 3).

Table 3. Load Test Results for 200 Virtual Users (VU)

Transaction/ Event	Execution for no of concurrent VU.	Avg Response Time (ms)	90% Line (ms)	Error %	SLA (ms)
TS001_Launch Site	200	257	293	0.00%	2000
TS002_Create Account	200	341	390	0.00%	2000
TS003_Add Details (Submit)	200	264	306	0.00%	2000
TS004_Logout	200	245	284	0.00%	2000

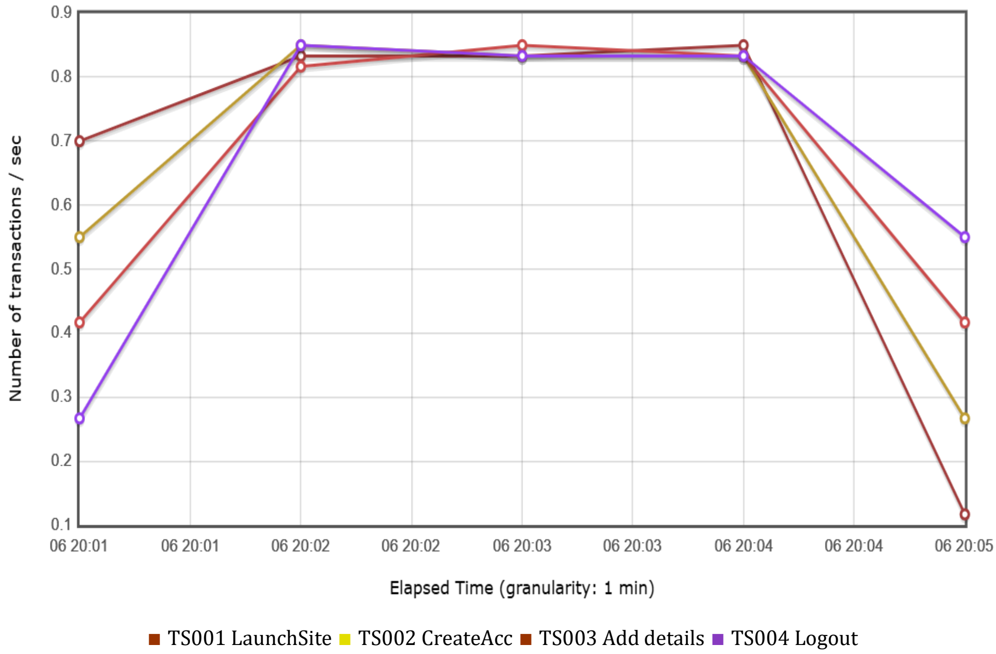

Figure 7. Throughput: Transaction per second-200 VU

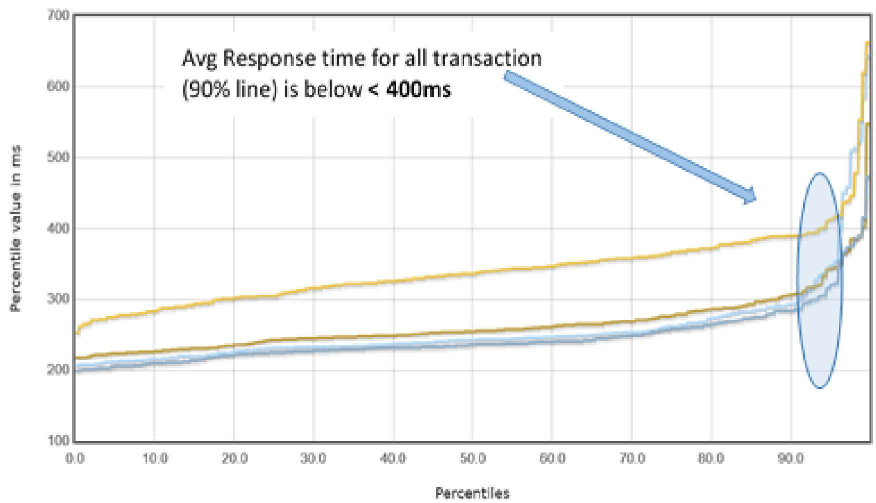

Figure 8. Average response time for 200 VU

The system demonstrated strong performance under an initial load of 200VU, with an average response time toggling around 250ms-300ms, against each transaction. With no request drop (i.e. error %) and 90% Line between 293-400ms.

The Stability of the system can be observed with throughput indicator of the system, it is observed against number of transactions per second (i.e. TPS). Moreover, it is seen that for all four (04) transactions the system is responding with 0.85-1.00 TPS during full execution of 200VU as shown in Fig.7. The system found to be in control with 1 CPU execution and 1 TB of memory in utilization. The SLA benchmark plays important role to understand the throughput insights, Avg Response time for 90%-line vs SLA.

Figure 8. Shows the 90% of the response time is below the 400ms against all transactions. For example, the 90% of observed response time against TS001_Launch site event is around 293ms throughout the test (Table 1), 390 ms for 'TS002_Create account' event, and 306ms and 284ms for 'TS003_Add Details' and 'TS004_Logout' event respectively.

This collectively shows that the response time is acceptably below the authors SLA threshold of 2000 ms for all transactions (Table 3) indicating the system's ability to handle concurrent operations efficiently.

5.1.2 Test at 300 Virtual Users

Post 200VU test observation, authors executed another test with simulated user load increased to 300 concurrent virtual users. The observed performance metrics displayed minor but noticeable change. Under a higher load of 300 virtual users, it is observed that the system-maintained average response time around 250-300 ms except the 'TS002_Create account' event which takes 420ms (Table 4). The system could able to execute 298 VU (out of 300VU) request against TS002 and TS003 transactions that show 0.33% of error.

As throughput is observed between 0.91-1.0 TPS for all four transactions as shown in Figure 9, it shows the stable behaviour of the system. To understand the system stability in depth- it is found that SLA compliance for all transactions, with a marginal increase (264ms -420ms) in average response time. The 90% average response is maintained as similar to <400ms against 300VU setup (Figure 10). The 90% of response time for TS001 was 253ms, wherein 380ms for TS002 and 287ms and 283ms for TS003 and TS004 transactions, respectively. Overall the results indicate that the system is scalable and capable of supporting increased user loads while maintaining error rates below acceptable limits.

Table 4. Load Test Results for 300 Virtual Users

Transaction/Event	Execution	Avg Response Time (ms)	90% Line (ms)	Error %	SLA (ms)
TS001_Launch Site	300	264	253	0.00%	2000
TS002_Create Account	298	420	380	0.33%	2000
TS003_Add Details (Submit)	298	279	287	0.33%	2000
TS004_Logout	300	246	283	0.00%	2000

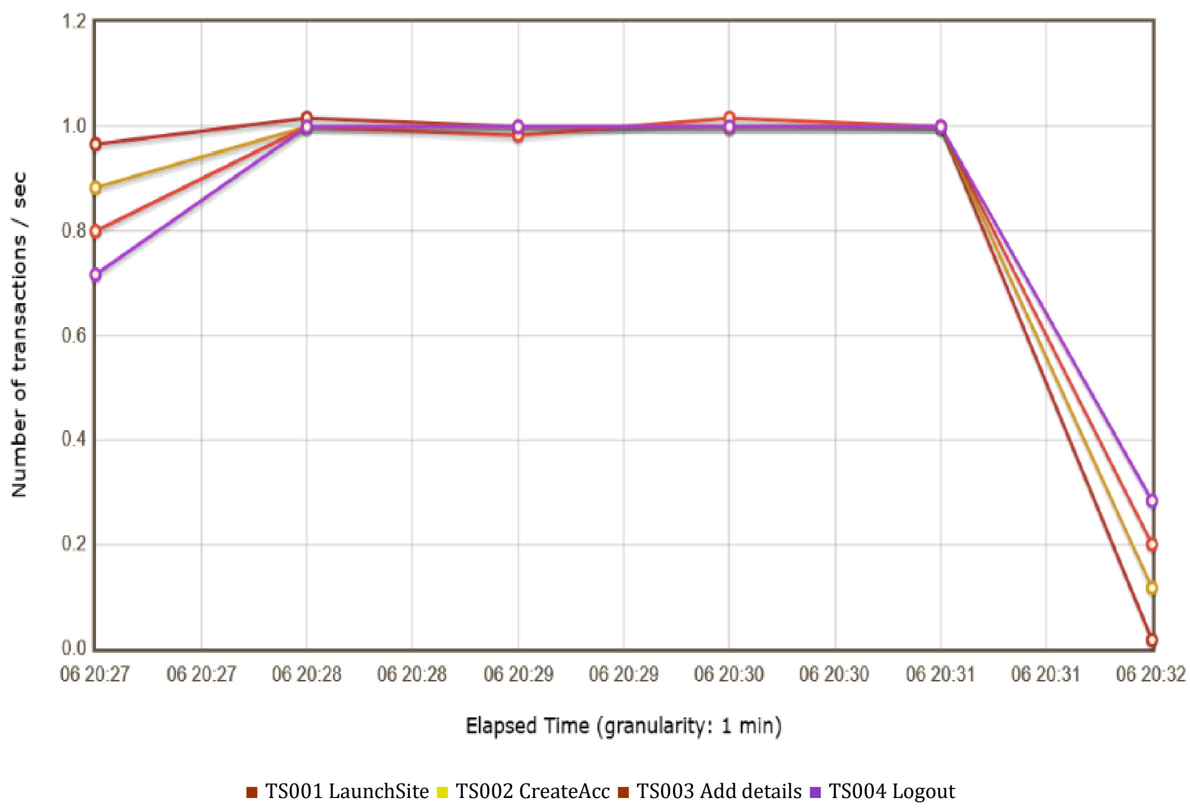

Figure 9. Throughput: Transaction per second-300 VU

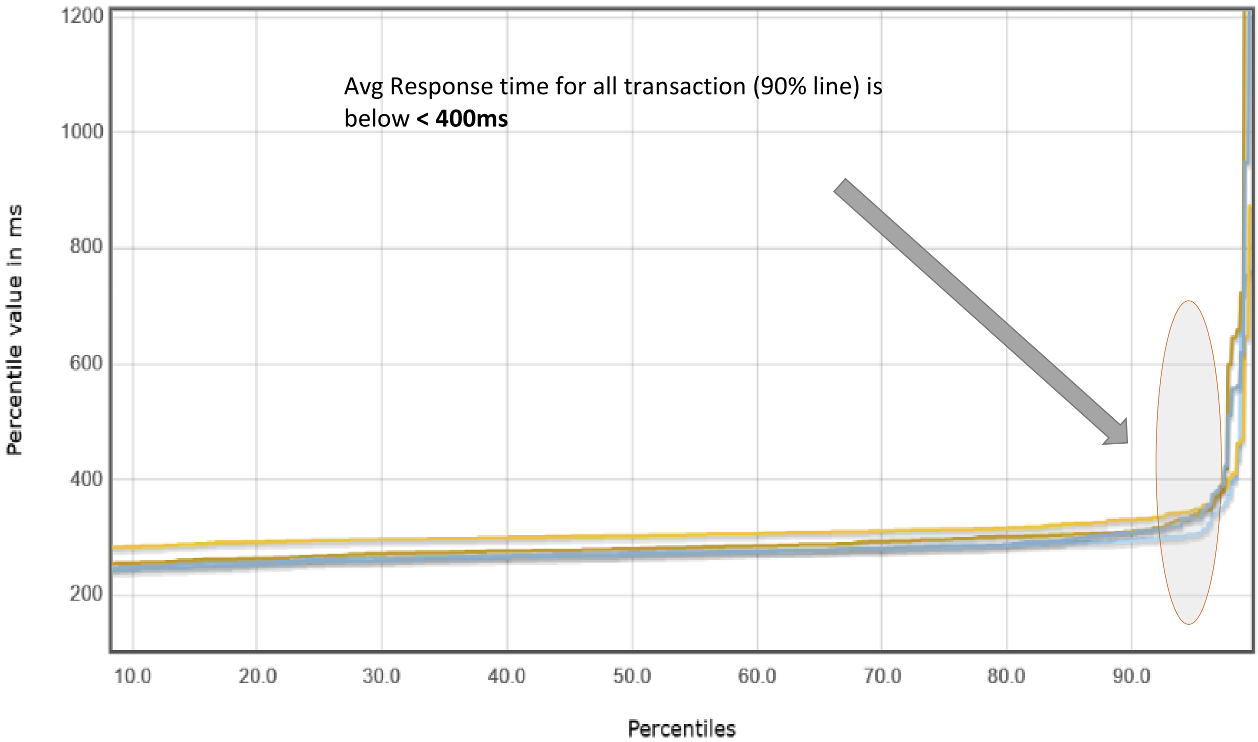

Figure 10. Average response time for 300 virtual users

6. Analysis and Discussion

After the execution of load test for different users set of 200VU and 300 VU scenario, it is seen that total four transactions all transactions response was executed successfully with ideal SLA benchmark set at 2000ms. As shown in Figure 11, it is seen that the all requests responses were executed successfully against 200 VU. The total 800 executions were made, wherein 200 execution requests comprising each of 4 transactions (i.e. 200 x 4 = 800) and two such iterations are carried out (i.e. 800 x 2 = 1600 requests response were under observations). It is seen that the 1596 request response were successfully executed within 400ms, and 04 request responses took more than 500ms but less than 1500ms. No request response took more than 1500ms. This shows the efficiency in response time of system under 200VU load.

On the same line as under 300VU conditions, total 2400 request's response (2 iterations) were under observation (300 x 4=1200x2 iterations = 2400) and it is seen that only 02 request were dropped i.e. error made against 2400 request executions. However, 2380 request's responses were executed under 500ms, 12 request having response time more than 500ms but less that 1500ms, and 06 requests took more that 1500ms response time as shown in Figure 12.

This shows the satisfactory response of the system behaviour under both load conditions and the archived stability with very minimal error at 300VU.

The throughput of the system was consistent across both scenarios, demonstrating a gradual increase in hits per second as user load increased (Table 5 and Figure. 13). The balanced throughput refers to suitably optimal indicator of the system stability. This stability underscores the framework's efficient resource management and capacity to handle concurrent requests without degradation in performance, as no error found against 200VU and minor against 300VU with average response time below SLA for all transactions.

Considering above all, the results confirm the effectiveness of the MAMATA CPS architecture in satisfying current maternal e-health system requirements. The key findings include:

- ✓ Speedy Response Times: Even with heavy user loads, the framework persistently kept response times below SLA levels to provide a flawless user experience.
- ✓ Scalability: The system showed great scalability, handling a maximum of 300 virtual users with no perceptible performance loss.
- ✓ Enhanced Security: Error rates were found to be very small, and the overall system reliability was increased by post-deployment security measures.

The outcomes highlight ability of MAMATA CPS framework to address the unique challenges of maternal e-healthcare system, particularly in rural and resource limited settings. The framework establishes a standard for upcoming e-health system designs by incorporating cutting-edge technologies and giving performance measurements a priority.

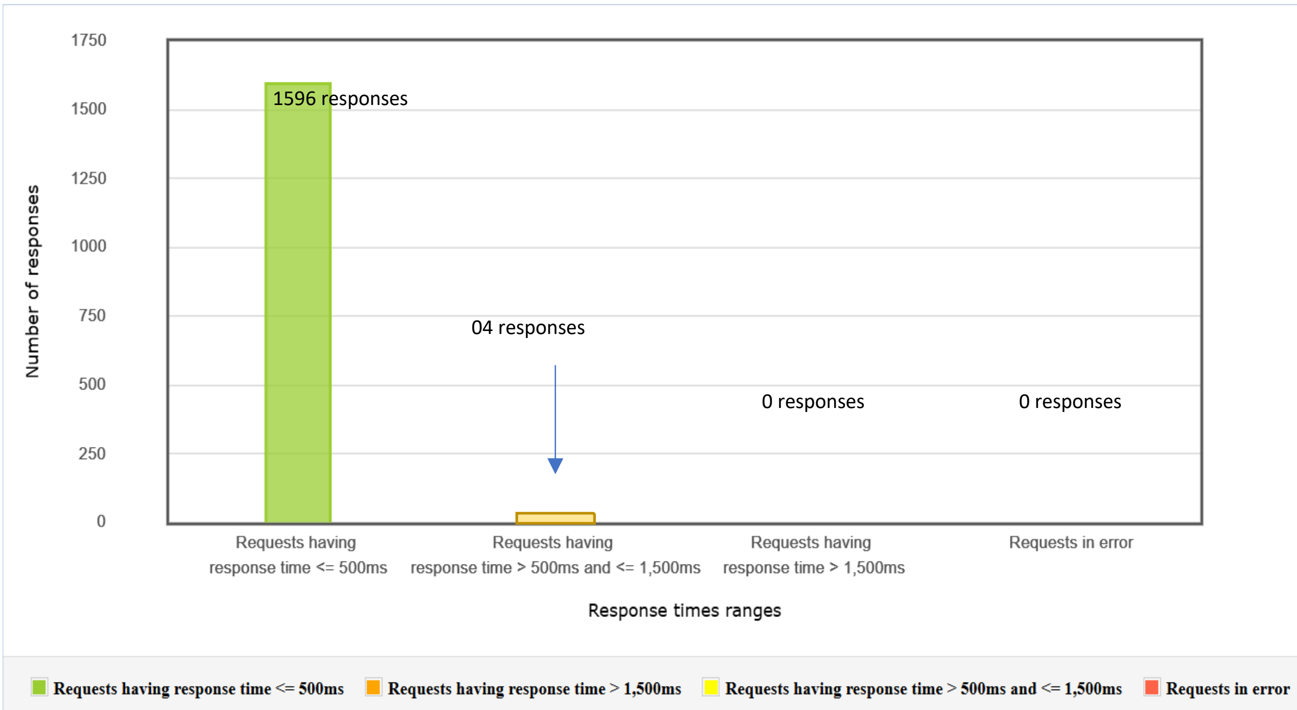

Figure 11. Response time overview against no of request

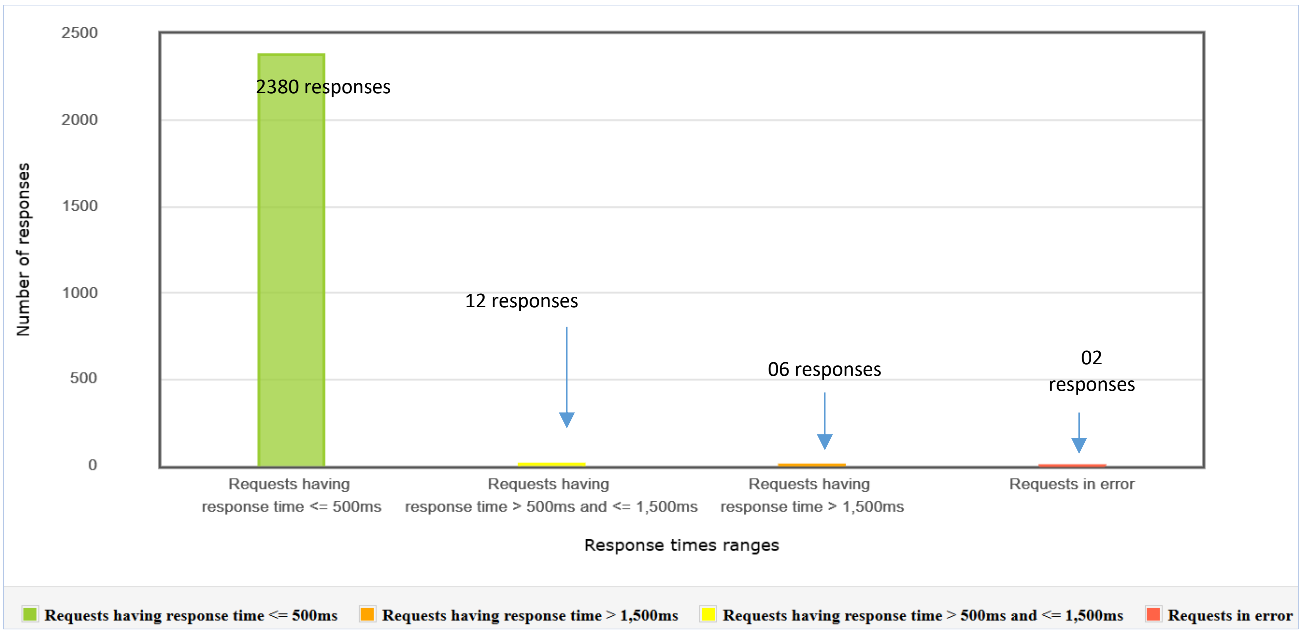

Figure 12. Response time overview against no of request

Table 5. Summary-Throughput (SLA vs Avg response tome) for 200 and 300 Virtual Users

Transaction/Event Name	**Average Response Time (in ms)**		**Errors in %**		**SLA Vs Res Time**
	200 VU	300 VU	200 VU	300VU	RT < SLA time
TS001_Launch_Site	**257**	**264**	0	0	**ACCEPTABLE**
TS002_Create_Account	**341**	**420**	0	**0.33%**	**ACCEPTABLE**
TS003_Additional_Details Submit	**264**	**279**	0	**0.33%**	**ACCEPTABLE**
TS004_Logout	**245**	**246**	0	0	**ACCEPTABLE**

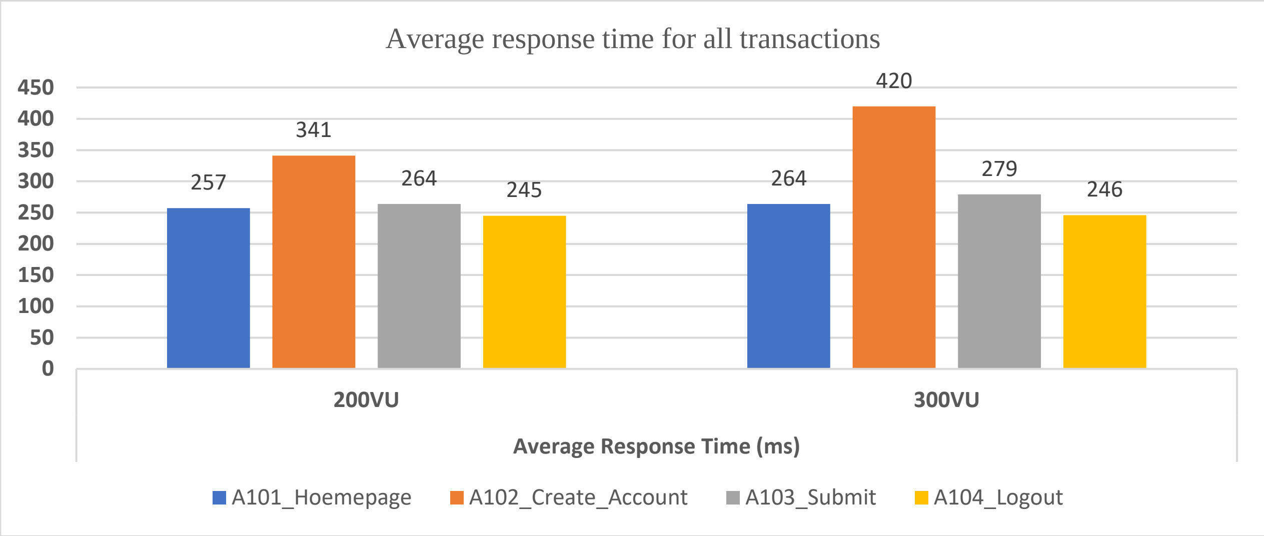

Figure 13. Average response/throughput analysis for 200 and 300 virtual users

7. Conclusion and Future Work

In terms of digitizing maternal e-healthcare, the MAMATA Cyber-Physical System (CPS) platform represents a significant advancement. This platform addresses challenges pertaining to performance, scalability, security, and user experience to address the concerns. Through the utilization of Laravel's robust and modular architecture, the framework combines essential components such as patient governance, data integration, maternity care, telemedicine, and data analytics. This results in the production of a comprehensive solution that is specifically built for maternal healthcare. In order to guarantee a low response time, rapid processing, and SLA compliance, even when there are an excessive number of transaction requests, the system has been subjected to extensive testing. To ensure the safety of the administration of data; encryption of data, verification procedures, and role-based access control are implemented. Furthermore, the uniformity of health records, the promotion of accessibility, and the functioning of the system in a smooth manner are all promoted by a user interface that is easy.

8. Future work

The framework's development will prioritize expanding its functionality through the integration of advanced technologies, such as Blockchain in healthcare, machine learning and artificial intelligence aimed at enhancing secure and accelerated clinical decision mechanism and distinctive ehealth service. Additional initiatives will aim to expand the role of ehealth with offline access for rural areas with multilingual support, which are critical in maternal care. for 'Alzheimer care' service for elderly patients, paediatric care, and neuro services. Looking towards the enriched data handling and response time, the MAMATA CPS framework will develop an optimal benchmark for maternal care service, especially, in rural demography where the electronics and other health infrastructure for maternal clinics are limited.

References

[1] M.I.M. Salleh, R. Abdullah, N. Zakaria, Evaluating the effects of electronic health records system adoption on the performance of Malaysian health care providers. BMC Med Inform Decis Mak 21, (2021)1-13. https://doi.org/10.1186/s12911-021-01447-4

[2] A. Raut, C. Yarbrough, V. Singh, B. Gauchan, D. Citrin, V. Verma, J. Hawley, A.H.Bangura, B. Shrestha, R. Schwarz, M. Adhikari, D.Maru, Design and implementation of an affordable, public sector electronic medical record in rural Nepal. Journal of innovation in health informatics, 24(2), (2017) 862. https://doi.org/10.14236/jhi.v24i2.862

[3] N. Kasoju, N.S. Remya, R. Sasi, S. Sujesh, B. Soman, C. Kesavadas, C.V. Muraleedharan, P.R. Harikrishna Varma, S. Behari, Digital health: trends, opportunities and challenges in medical devices, pharma and bio-technology. CSI Transactions on ICT, 11(1), (2023) 11-30. https://doi.org/10.1007/s40012-023-00380-3

[4] J. Al-Jaroodi, N. Mohamed, E. Abukhousa, Health 4.0: On the Way to Realizing the Healthcare of the Future. IEEE access, 8, (2020) 211189–211210. https://doi.org/10.1109/ACCESS.2020.3038858

[5] Y. Yu, S. Yao, K. Wang, Y. Chen, F. Wang, Y. Lin, (2023) Pursuing Equilibrium of Medical Resources via Data Empowerment in Parallel Healthcare System. IEEE International

Conference on Systems, Man, and Cybernetics (SMC), IEEE, USA. https://doi.org/10.1109/SMC53992.2023.10394360

[6] S. Abbasi; B. K. Jung, (2023) Overview and Insight: Performance of Digital Health Systems during the COVID-19 Pandemic. IEEE.

[7] S.E. Kafhali, K. Salah, Performance modelling and analysis of Internet of Things enabled healthcare monitoring systems. IET Networks, 8(1), (2019) 48-58. https://doi.org/10.1049/iet-net.2018.5067

[8] P.C. Dhage, R.A. Thakker, K.K. Warhade, Security Mechanism in MAMATA Healthcare System Using Rule based Algorithm for Maternal Hospitals and Pathology Laboratories. Journal of Internet Services and Information Security (JISIS), 14(4), (2024) 292-311.

[9] N.K. Abujbara, J. A. Worley, Performance measurement indicators in the healthcare industry: a systematic review. Global Business and Economics Review, 21(1), (2019) 43– 68. https://doi.org/10.1504/GBER.2019.096857

[10] K.D. Larson, The role of service level agreements in IT service delivery. Information Management & Computer Security, 6(3), (1998) 128-132. https://doi.org/10.1108/09685229810225029

[11] R. Sivan, Z.A. Zukarnain, Security and Privacy in Cloud-Based E-Health System. Symmetry, 13(5), (2021) 742. https://doi.org/10.3390/sym13050742

[12] I.K. Ayu, N. Nuryasin, Usability Analysis of the Health Center Management Information System (Simpus) Using the Heuristic Evaluation Method at Ajibarang 1 Health Center. Applied Information System and Management (AISM), 2(2), (2019) 51-56. https://doi.org/10.15408/aism.v2i2.20085

[13] N. Verma, B. Mamlin, J. Flowers, S. Acharya, A. Labrique, T. Cullen, OpenMRS as a global good: Impact, opportunities, challenges, and lessons learned from fifteen years of implementation. International journal of medical informatics, 149, (2021) 104405. https://doi.org/10.1016/j.ijmedinf.2021.104405

[14] BahmniTM Open Source EMR | ThoughtWorks, BahmniTM. https://www.bahmni.org/

[15] B. C. Study, “Babylon Case Study,” Kubernetes, Sep. 06, 2020. https://kubernetes.io/case-studies/babylon

[16] P. Thakur, P. Jadon, (2023) Django: Developing web using Python. IEEE- 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Noida, India. https://doi.org/10.1109/ICACITE57410.2023.10183246

[17] K.M. Kuo, C.F. Liu, P.C. Talley, S.Y. Pan, Strategic Improvement for Quality and Satisfaction of Hospital Information Systems. Journal of Healthcare Engineering, 2018(1), (2018) 3689618. https://doi.org/10.1155/2018/3689618

[18] H. Mei, B.J. van Beijnum, V. Jones, H. Hermens, (2008) Enhancing the performance of mobile healthcare systems based on task-redistribution. In IEEE INFOCOM Workshops 2008, IEEE, USA. https://doi.org/10.1109/INFOCOM.2008.4544603

[19] S. Purbey, K. Mukherjee, C. Bhar, Performance measurement system for healthcare processes. International Journal of Productivity and Performance Management, 56(3), (2007) 241-251. https://doi.org/10.1108/17410400710731446

[20] M. Fridell, S. Edwin, J. von Schreeb, D.D. Saulnier, Health System Resilience: What Are We Talking About? A Scoping Review Mapping Characteristics and Keywords. International journal of health policy and management, 9(1), (2020) 6-16. https://doi.org/10.15171/ijhpm.2019.71

[21] G. Dicuonzo, G. Galeone, M. Shini, A. Massari, Towards the Use of Big Data in Healthcare: A Literature Review. Healthcare, 10(7), (2022) 1232. https://doi.org/10.3390/healthcare10071232

[22] L.R. Kalankesh, R. Fein, R. Fein, S. Damanabi, Factors Influencing User Satisfaction with Information Systems: A Systematic Review. Galen Medical Journal, 9, (2020) e1686. https://doi.org/10.31661/gmj.v9i.1686

[23] D.F. Sittig, D. González, H. Singh, Contingency planning for electronic health record-based care continuity: A survey of recommended practices. International Journal of Medical Informatics, 83(11), (2014) 797-804. https://doi.org/10.1016/j.ijmedinf.2014.07.007

[24] B. Neha, S.K. Panda, P.K. Sahu, D. Taniar, Energy and latency-balanced osmotic-offloading algorithm for healthcare systems. Internet of Things, 26, (2024) 101176. https://doi.org/10.1016/j.iot.2024.101176

[25] X. Li, K. You, Real-time tracking and detection of patient conditions in the intelligent m-Health monitoring system. Frontiers in Public Health, 10, (2022) 2296-2565. https://doi.org/10.3389/fpubh.2022.922718

[26] J. Chen, Y. Wang, F. Magrabi, (2017) Downtime in digital hospitals: an analysis of patterns and causes over 33 months. In Integrating and Connecting Care, IOS Press. https://doi.org/10.3233/978-1-61499-783-2-14

[27] C. Pou-Prom, J. Murray, S. Kuzulugil, M. Mamdani, and A.A. Verma, from compute to care: lessons learned from deploying an early warning

system into clinical practice. Frontiers in Digital Health, 4, (2022) 932123. https://doi.org/10.3389/fdgth.2022.932123

[28] N.C. Nelson, Downtime procedures for a clinical information system: a critical issue. Journal of critical care, 22(1), (2007) 45-50.

[29] E.P. Larsen, A.H. Rao, F. Sasangohar, Understanding the scope of downtime threats: A scoping review of downtime-focused literature and news media. Health Informatics Journal, 26(4), (2020) 2660-2672. https://doi.org/10.1177/1460458220918539

[30] G.E. Weissman, A. Crane-Droesch, C. Chivers, T. Luong, A. Hanish, M.Z. Levy, J. Lubken, M. Becker, M.E. Draugelis,, G.L. Anesi, P.J. Brennan, J.D. Christie, C.W. Hanson, M.E. Mikkelsen, S.D. Halpern, Locally informed simulation to predict hospital capacity needs during the COVID-19 pandemic. Annals of internal medicine, 173(1), (2020) 21-28. https://doi.org/10.7326/M20-1260

[31] E. Nazari, M. Aghemiri, S.M. Tabatabaei, S.M. Mostafavi, S. Aalaei, S.E. HasanAbady, H. Tabesh, Thesis Conducted By Student In Medical Informatics Field Based On Health Informatics Competencies Framework: Astudy In Iran. Frontiers in Health Informatics, 9(1), (2020) e42.

[32] H.M. Hamer, O.I.A. Ashour, M. Ilyas, A web application for company reports management system. In Proceedings of the 6th International Conference on Engineering & MIS, (2020) 1-5. https://doi.org/10.1145/3410352.3410815

[33] E. Sabna, D. Yasriady, M. Giatman, Attendance Management Based on MVC and Laravel in the Municipal Government of Pekanbaru. Jurnal Pendidikan Teknologi Kejuruan, 4(1), (2021) 1. https://doi.org/10.24036/jptk.v4i1.16423

[34] A.A. Prayogi, M. Niswar, I. Indrabayu, M.R. Rijal, Design and Implementation of REST API for Academic Information System. IOP Conference Series Materials Science and Engineering, 875(1), (2020) 12047. https://doi.org/10.1088/1757-899X/875/1/012047

[35] P.C. Dhage, R.A. Thakker, K.K. Warhade, "Novel mamata healthcare cyber physical system for implementation and integration of rural maternal treatment practices in maternal hospital units and pathology labs" South Africa, ZA2022/04698A, 28-04-2022.

[36] M. Aydos, Ç. Aldan, E. Coşkun, A. Soydan, Security testing of web applications: A systematic mapping of the literature. Journal of King Saud University-Computer and Information Sciences, 34(9), (2022) 6775-6792. https://doi.org/10.1016/j.jksuci.2021.09.018

[37] B.P. Mramba, S.F. Kaijage, Design of an Interactive Mobile Application for Maternal, Neonatal and Infant Care Support for Tanzania. Journal of Software Engineering and Applications, 11, (2018) 569-584. https://doi.org/10.4236/jsea.2018.1112034

[38] Pradeepkumar Dhage et. al, Copyright No: L-101275/2021 (India), 1187029 (Canada), dated 05-04-2021, 06-10-2021 respectively.

[39] R.T. Sutton, D. Pincock, D.C. Baumgart, D. Sadowski, R.N. Fedorak, K.I. Kroeker, An overview of clinical decision support systems: benefits, risks, and strategies for success. npj Digital Medicine, 3(1), (2020). https://doi.org/10.1038/s41746-020-0221-y

[40] H.C. Taylor, Editorial: Maternity Care Monitoring. International Journal of Gynecology & Obstetrics, 17(1), (1979) 23–23. https://doi.org/10.1002/j.1879-3479.1979.tb00111.x

[41] M. Stram, T. Gigliotti, D. Hartman, A. Pitkus, S.M. Huff, M. Riben, Walter H. Henricks, MD; Navid Farahani, Pantanowitz, L. (2020). Logical observation identifiers names and codes for laboratorians: potential solutions and challenges for interoperability. Archives of pathology & laboratory medicine, 144(2), 229-239. https://doi.org/10.5858/arpa.2018-0477-RA

[42] Dataset, "Entrust is a succinct and flexible way to add Role-based Permissions to Laravel 7", (2020).

[43] K. Ekandem, Caching in Laravel with Redis: a complete guide, Honeybadger Developer Blog, Aug. 29, 2024. https://www.honeybadger.io/blog/laravel-caching-redis/

[44] J. Wang, J. Wu, (2019) Research on Performance Automation Testing Technology Based on JMeter. International Conference on Robots & Intelligent System (ICRIS), IEEE, China. https://doi.org/10.1109/ICRIS.2019.00023

[45] M. Harchol-Balter, Open problems in queueing theory inspired by datacenter computing. Queueing Systems, 97, (2021) 3–37. https://doi.org/10.1007/s11134-020-09684-6

[46] A.K. Swain, V.R. Garza, Key factors in achieving Service Level Agreements (SLA) for Information Technology (IT) incident resolution. Information Systems Frontiers, 25(2), (2023) 819-834. https://doi.org/10.1007/s10796-022-10266-5

Authors Contribution Statement

Pradeepkumar C Dhage: Conceptualization, Methodology, Writing- Original Draft. R.A Thakker: Data Curation, Investigation, Writing - Review & Editing. Krishna K Warahde: Formal Analysis, Visualization, Supervision.

Has this article screened for similarity?

Yes

Funding

The authors declare that no funds, grants or any other support were received during the preparation of this manuscript.

Competing Interests

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Electronic Supplementary Information

Implementations code has been included.

Data Availability

The data supporting the findings of this study can be obtained from the corresponding author upon reasonable request.

About the License