iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

SYSTEM ENGINEERING PROCESS METHODOLOGY FOR DEVELOPING CLINICAL GROUP DECISION SUPPORT SYSTEM IN DENGUE MANAGEMENT

Muhammad Syaukani [1]
Fakultas Ilmu Komputer, Institut Teknologi Bisnis dan Bahasa 'Dian Cipta Cendikia'
Bandar Lampung, Indonesia
*Corresponding Author : mbsyaukani@gmail.com

ABSTRACT
The development of the Clinical Group Decision Support System (CGDSS), so far, generally combines traditional and modern methods, such as Waterfall, Agile, and Prototyping. Challenges in developing CGDSS are data integration issues, user complexity, compliance with clinical standards, data security, and complexity in designing system interfaces and functionality. In addition, the algorithm's accuracy in providing data-based recommendations is also vital because errors can directly impact patient care. This study developed CGDSS for Dengue management with the System Engineering Process Methodology (SEPM) approach as a solution to combining the advantages of various methods by emphasizing structured, flexible, and adaptive system engineering. It can overcome various problems related to data integration, algorithm accuracy, and user acceptance.

Keywords:
CGDSS, SEMP, Dengue

INTRODUCTION
A decision support system (DSS) is an information system that helps individuals or organizations make decisions by utilizing data, analysis, and predictive models. The advantage of DSS is its ability to process complex data into information that can improve the accuracy and speed of decisions. DSS is helpful in planning, evaluation, and problem-solving. However, its weakness is its dependence on the quality of the data and models used; if the data or model is poor, the decision results can also be wrong. The primary function of DSS is to support complex decisions, both at the strategic and operational levels (Turban, 2021). Meanwhile, Group Decision Support System (GDSS) is a development of DSS that supports group decision-making by facilitating communication, coordination, and collaboration between team members in a virtual environment. The advantage is that it supports inclusive decision-making by considering various points of view. The disadvantage is that coordination is time-consuming, and the a need for technological understanding from all team members. The primary function of GDSS is to help teams evaluate options, discuss, and choose optimal decisions collaboratively (Melin, 2022).

Clinical Decision Support System (CDSS) is a system that assists healthcare professionals in clinical decision-making by analyzing patient data and medical guidelines, providing evidence-based recommendations, and reducing diagnostic errors. However, CDSS relies on algorithms that can only be safe if they are accurate and up-to-date (Kawamoto, 2021). Meanwhile, the Clinical Group Decision Support System (CGDSS) combines the concepts of GDSS and CDSS to support medical team collaboration, utilizing clinical data analysis and predictive algorithms for fast and precise recommendations. Its disadvantages include complex system integration and potential data misinterpretation, with the primary function of assisting the medical team in more effective case coordination, diagnosis, treatment, and planning (Zhang, 2021).

These four systems, DSS, GDSS, CDSS, and CGDSS, have their advantages and functions for different contexts: DSS supports data-driven decisions, GDSS for team collaboration, CDSS assists in clinical contexts, and CGDSS integrates team collaboration with clinical analysis for medical case management.

The development and implementation of CGDSS for Dengue management face challenges in integrating data from various sources, such as electronic medical records, laboratory results, and epidemiological data, which require complex coordination and standards.

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

In addition, the diversity of user needs, such as doctors, nurses, and other medical personnel, adds complexity to designing the system interface and functionality. The accuracy of the algorithm in providing data-based recommendations is also important because errors can have a direct impact on patient care.

User acceptance of the system, such as resistance to change and complexity of use, is an important constraint. In addition, data security aspects must be considered to protect patient health information according to applicable regulations, such as GDPR or HIPAA. A methodological and structured approach is needed to address these issues.

DSS development methodologies combine traditional and modern methods, such as Waterfall, Agile, and Prototyping. Although each has advantages, these approaches are often less integrated or need to be more systematic. Therefore, a System Engineering Process Methodology (SEPM)-based approach is required to combine the advantages of these various methods, emphasizing structured system engineering and involving all stakeholders throughout the development process. (Martin, 2021).

This approach plays a vital role in overcoming various obstacles that may arise in developing and implementing CGDSS, such as data integration issues, user complexity, compliance with clinical standards, and data security. Using SEPM, CGDSS development will be more structured, flexible, and adaptive and can overcome various issues related to data integration, algorithm accuracy, and user acceptance. SEPM allows the development team to collaborate with the medical team and stakeholders at every stage of development to ensure successful implementation and maximum benefits for Dengue Fever management. This study develops CGDSS in dengue management using systems engineering process methodology.

LITERATURE REVIEW

Research conducted by Manuel Mora (2011) proposed IDSSE-M, a software systems engineering-based methodology for building intelligent Decision-Making Support Systems (DMSS), overcoming the limitations of traditional methods such as Waterfall and Prototyping. IDSSE-M adopts an evolutionary-incremental approach with four main phases: Project Initiation, System Design, Development and Evaluation, and Project Completion. This methodology extends the paradigm of Turban and Aronson (1998) and the method of Saxena (1991) and has proven effective in improving the quality of DMSS development. User evaluations show that IDSSE-M is easy to use, improves quality, and provides practical benefits. This methodology has also been applied in universities with the results of more than 30 DMSS prototypes by postgraduate students.

Furthermore, S. Vallance's research (2011) discusses the challenges in decision-making in managing new product development (NPD), often faced with high uncertainty and complexity. Wrong decisions can impact time, quality, and resource allocation. Decision Support Systems (DMSS) can reduce risk but are often poorly integrated with the organizational context. This study evaluates various system development methodologies (DSMs) based on criteria such as business needs, organizational culture, and human factors. Key findings suggest that no single DSS fully addresses the needs of DMSS development, with gaps primarily in neglecting organizational culture and human factors. This study emphasizes the importance of a holistic approach that encompasses both technical and non-technical factors to ensure solutions are relevant and accepted by the organization.

Huweida A (2022) proposed a structured approach to analyzing and designing Decision Support Systems (DSS), which aims to assist complex decision-making in organizations. The developed framework consists of four stages: (1) System Problem Definition, (2) Problem Domain Understanding, (3) System Analysis, and (4) System Design. The main focus is on Knowledge-driven DSS, which is the core of this approach. This model provides clear guidance for developers in designing DSS that suit their needs and integrates decision-making methods with systems engineering techniques. This study emphasizes the importance of DSS in decision-making and suggests further studies to test its application in real case studies.

Svetlana Lawrence (2024) developed a Model-Based Systems Engineering (MBSE) framework to support strategic decision-making in developing clean energy solutions, such as hydrogen, to decarbonize the energy sector. The method combines systems thinking and model-based systems engineering with MBSE tools such as Innoslate and multi-criteria decision analysis (MCDA). The research findings show that MBSE enables comprehensive evaluation and supports the modeling and simulating of hydrogen technologies, such as SMR-CCS, low-temperature electrolysis (LTE), and high-temperature electrolysis (HTSE). This approach accelerates the decision-making process, reduces the risk of errors, and improves tracking and change management. In conclusion, MBSE is effective for decision-making in complex energy systems, and the developed framework

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

provides a systematic guide to evaluating innovative solutions. However, further studies are needed to determine its implementation on an industrial scale.

Verma (2021) reviews the application of SEPM in DSS development, combining an iterative approach with systems engineering principles. The author identifies the benefits of using SEPM to improve stakeholder collaboration and ensure that business, technical, and operational requirements are well integrated throughout the DSS lifecycle. Henshaw (2022) explores how a SEPM-based approach can improve DSS development by leveraging an iterative model, improving communication flows between teams, and ensuring the involvement of all stakeholders at each stage of development.

Smith (2020) discusses the challenges and solutions of applying System Engineering Process Methodology (SEPM) in developing DSS in real-world environments. The author argues that although traditional methods are often used, integration with SEPM enhances the ability of DSS to handle more complex problems with a more structured approach. Jackson (2023) examines the application of the SEPM method, which combines various systems engineering techniques in developing DSS that can handle high uncertainty and complexity and improve decision accuracy in organizations. Of the several studies conducted previously, none have discussed the application of System Engineering Process Methodology for Developing Clinical Group Decision Support System in Dengue Management.

METHODOLOGY

The system engineering process methodology for developing the Clinical Group Decision Support System includes the main phases: Requirements Analysis, System Design, Implementation, Testing, Deployment, and Maintenance. These phases are complemented by visual elements relevant to technology and health, as can be seen in the following image.

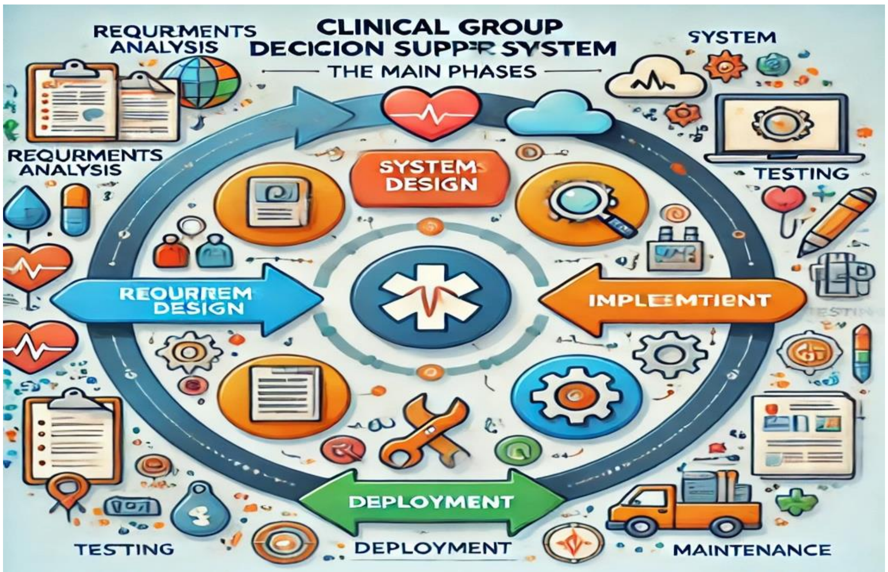

Figure 1. Stages of System Engineering Methodology for CGDSS

The following is an explanation of the stages in the Systems Engineering Process Methodology (Sage, 2020) to develop a Clinical Group Decision Support System (CGDSS):

1. Requirements Analysis

   Requirements Analysis is a systematic process aimed at identifying, understanding, and documenting the needs or requirements of stakeholders in a project or system. This process involves gathering requirements from various sources, such as interviews, surveys, or observations, and analyzing them in depth to ensure that the requirements are clear, consistent, and implementable. Requirements Analysis also involves classifying requirements into functional, which describe what the system must do, and non-functional, which cover quality aspects such as performance or security. The results of this process are usually compiled in a formal document, such as a Software Requirements Specification (SRS), which forms the basis for the

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

design and development of the solution. Ensuring that requirements are understood from the start helps reduce the risk of errors, saves costs, and ensures that the result meets stakeholder expectations.

2. System Design
   System Design is the process of planning and organizing the elements of a system to create a solution that meets the needs identified in the analysis phase. This process includes the selection of system architecture, component design, interfaces, and working mechanisms and how these elements interact to achieve the desired goals. System design also considers technical aspects such as scalability, security, efficiency, and reliability. The system design document guides the development team in building, implementing, and integrating the system. With a structured approach, System Design ensures that the designed solution meets functional needs and functions effectively in the specified operational environment.
3. Implementation
   Implementation is a stage in the system or software development cycle where the design that has been made in the design stage is applied in a tangible form, such as program code, hardware configuration, or system integration. This process involves developing, initial testing, and integrating system components to ensure that everything runs according to specifications. In addition, implementation includes applying the system to the operational environment, including installation, data migration, user training, and technical documentation. This stage focuses on transforming the design into a solution that stakeholders can use effectively by ensuring that the system functions according to the needs that have been set.
4. Testing
   Testing evaluates a system or software to ensure that the developed solution functions according to the requirements specifications and is free from errors or defects. This process includes various types of testing, such as functional testing to verify that each feature works as expected, and non-functional testing to evaluate aspects such as performance, security, and reliability. Testing is carried out systematically through various levels, starting from unit testing (testing individual components), integration testing (testing interactions between components), system testing (testing the entire system), and user acceptance testing (testing by end users). The primary purpose of testing is to identify and fix problems as early as possible, ensure high quality, and increase user satisfaction with the resulting system.
5. Deployment
   Deployment is implementing a system or software developed into an operational environment, so end users can use it. This stage includes various activities, such as software installation, system configuration, data migration from the old system to the new one, and final testing to ensure the system runs well in the production environment. Deployment also involves user training, guidance documentation, and providing technical support to ensure a smooth transition. This process can be done in several approaches, such as significant bang deployment (all at once) or stages, depending on the system's complexity and the organization's needs. The main goal of deployment is to ensure that the system is ready to use and can meet the business and technical needs that have been determined.
6. Maintenance
   Maintenance is an ongoing process to ensure that a system or software that has been implemented continues to function correctly, meets user needs, and adapts to environmental changes or new requirements. This process includes various activities, such as fixing bugs discovered after implementation, enhancing features or functions to meet evolving needs, and adapting to changes in technology, regulations, or the operational environment. Maintenance also includes regular system monitoring to identify potential problems and ensure optimal performance. This stage aims to extend the system's life, increase user satisfaction, and minimize operational disruptions by keeping the system up-to-date and reliable.

RESULTS AND DISCUSSION

Based on preliminary research conducted by the author at Sultan Suriansyah Regional General Hospital, Banjarmasin City and by the Decree of the Minister of Health of the Republic of Indonesia Number HK.01.07/MENKES/4636/2021 concerning National Guidelines for Medical Services for the Management of Dengue Infections in Children and Adolescents, and Decree of the Minister of Health of the Republic of Indonesia Number HK.01.07/MENKES/9845/2020 concerning National Guidelines for Medical Services for the Management of Dengue Infections in Adults, the treatment of dengue infection patients is carried out starting from the presumptive identification process during screening to the working diagnosis. The presumptive identification process during screening is determined by several medical personnel such as nurses, co-ass

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

doctors, and general practitioners. At the same time, internal medicine specialists and clinical pathology specialists carry out the working diagnosis process.

Emergency room staff select patients according to their emergency conditions as the priority of patient service by the provisions for emergency patient services that apply and are not based on the order of patient arrival, then sort patients based on their therapeutic needs and available resources. One of the indicators of the minimum service standards for hospitals in the emergency room is the response time, which is the standard time of 5 minutes for patients to be served when they arrive at the emergency room. In the service process in the emergency room, there are still cases of dengue infection patients who are not handled quickly and appropriately because the doctor on duty is not there and the limited knowledge of a nurse; in addition, as many as 11 percent of patients are not handled quickly within 1-4 hours when the dengue infection case occurs.

When the patient arrives at the ER, they are registered with their identity, such as name, date of birth, and address, and then triage is continued, which is the screening process. Screening can be defined as a presumptive identification of disease, a series of activities to conduct an initial emergency assessment on every patient who comes to the ER; in this case, patient screening is carried out at the beginning of the primary triage, which also includes diagnostic methods such as anamnesis (interview of disease history) and physical examination.

Screening is carried out by medical personnel such as general practitioners, co-assistant doctors, and nurses based on their experience by conducting presumptive identification of a symptom such as a symptom of dengue infection. The result of presumptive identification is the disease category, namely severe or mild dengue. If mild dengue is identified, the patient is admitted for outpatient care. In contrast, if severe dengue is identified, the consultation process is continued with a specialist doctor, such as an internal medicine specialist and a clinical pathology specialist, to carry out a working diagnosis. Based on the results of the working diagnosis, a diagnosis is determined. Namely, if the patient is declared severe dengue, then medical action is carried out by a specialist doctor to determine whether the patient is hospitalized/ICU and the therapy. The following is a picture of the dengue patient management process.

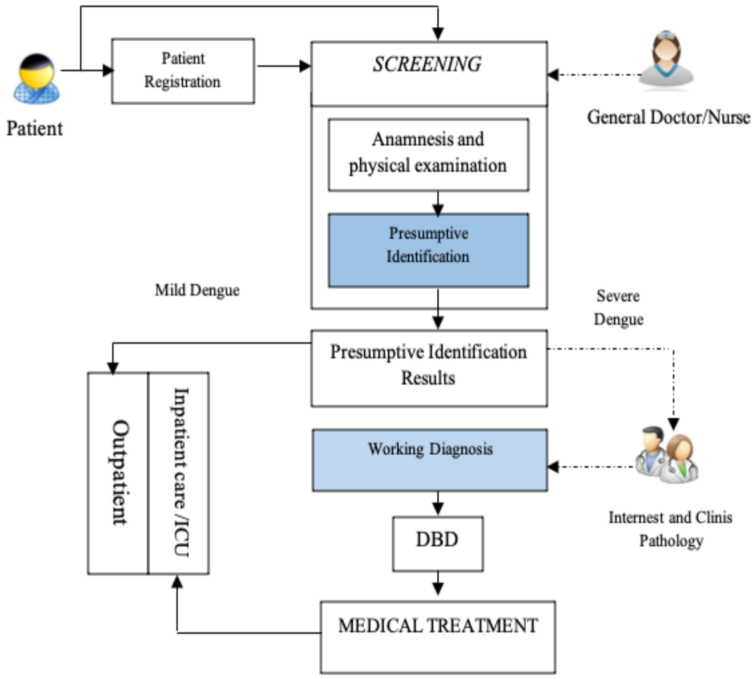

Figure 2. Management of Dengue Patients

Based on the above, a tool is needed that can utilize a computer-based system, namely a media that can help medical personnel in carrying out work in the clinical field, namely creating a Clinical Group Decision Support System (CGDSS) for Dengue management with the System Engineering Process (SEP) methodology that requires comprehensive design. System Engineering Process (SEP) is a systematic approach to developing, operating, and maintaining a complex system. System Engineering Process (SEP) is a methodology that focuses

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

on in-depth analysis, modeling, and testing in the system development cycle, which aims to ensure that the system meets user needs and functions optimally. System Engineering Process (SEP) is perfect for projects that require coordination between elements and ensure that the system meets its technical and operational needs. This methodology involves several stages of understanding, designing, testing, and maintaining the system.

1. **Requirement Analysis**

   This stage aims to understand the system and organizational needs related to dengue management. The needs collected include technical, operational, and environmental needs and user limitations and expectations must be considered. The following of the System Requirement Specification - SRS) for the Clinical Group Decision Support System (CGDSS) for dengue management.
   - Objective: Identify a clinical group decision support system's user, clinical, and technical needs.
   - Steps:
   - Conduct interviews with clinical teams (physicians, nurses, specialists).
   - Collect data on clinical workflows and group needs.
   - Determine key features such as patient data management, decision analytics, and interoperability.
   - Outcome:
   - Clear and verifiable requirements specification document.

Based on the System Requirement Specification, the System Requirement Specification Document (SRS) for the Clinical Group Decision Support System (CGDSS) for Dengue Management was obtained as follows:

Table 1. System Requirements Specification Document

1. Introduction	1.1 Purpose This document defines the need for a CGDSS system to detect, diagnose, manage, and monitor dengue cases. The system aims to assist health workers in making evidence-based decisions to improve clinical outcomes. 1.2 Scope The system is designed to: 1. Detect potential dengue case clusters based on epidemiological data. 2. Provide recommendations for diagnosis and treatment according to medical clusters. 3. Integrate patient data from multiple sources, such as laboratories and electronic medical records (EMR). 4. Support predictive analytics to identify patterns of dengue transmission. 1.3 Stakeholders - Doctors: Need evidence-based information for diagnosis and treatment. - Nurses: Use the system for patient monitoring. - Public Health Managers: Monitor outbreak clusters and make policies. - System Developers: Ensure development is by specifications.
2. General System Description	2.1 Main Fuction of the system 1. Data Management: Collect patient data, laboratory results, and geographic information. 2. Clinical Decision Support: Provide recommendations based on medical protocols. 3. Outbreak Monitoring: Visualize the spread of dengue through epidemiological maps. 4. Reporting and Analysis: Provide case reports and trends for policy planning. 2.2 System Boundary

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

	- The system can only be accessed by verified users. - The system must be connected to the network infrastructure of the hospital or health center.
3. Fungsional Requirements	3.1 Data Management Module - The system must be able to import patient data from EMR. - The system must support laboratory data management, including NS1 and platelet tests. 3.2 Decision Support Module - The system must provide diagnostic recommendations based on the symptoms entered. - The system must provide treatment recommendations based on the severity of dengue (DHF/DSS). 3.3 Outbreak Monitoring Module - The system must visualize a geographic map of dengue spread based on patient location. - The system must provide notification if there is an increase in cases in an area. 3.4 Report and Analysis Module - The system must be able to generate weekly and monthly case reports. - The system must support trend analysis using historical data.
4. Non-Functional Needs	4.1 Performance - The system must process up to 10,000 patient data per day. - The system response for data search is a maximum of 2 seconds. 4.2 Security - The system must use two-factor authentication for access. - Patient data must be encrypted according to HIPAA standards. 4.3 Scalability - The system must be scalable to support additional hospitals. 4.4 User Interface - The interface must be user-friendly and accessible on desktops and tablets. - The language used must support local languages and English
5. System Flowchart	It displays the flow from patient data input, processing in CGDSS, and output in the form of recommendations or reports.
6. Software and Hardware Requirements	- Software: - Operation System: Windows/Linux - Database: PostgreSQL/MySQL - Framework: Django/Flask - Hardware: - Server with minimum specifications: 16-core CPU, 64GB RAM, 1TB SSD. - User terminal: PC or tablet with minimum 4GB RAM..
7. Success Criteria	- The system can provide diagnostic recommendations with an accuracy of >95%. - The system is integrated with EMR without any problems. - The system successfully detects case clusters with a sensitivity of >90%.
8. Testing and Validation Plan	- Functional Test: Testing all modules as needed. - Load Test: Measuring performance under peak conditions. - Clinical Test: Validating the system in a hospital with real cases

2. System Design

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

This stage focuses on the design of the CGDSS system architecture. The steps include:
- System architecture design: Determine system components such as data analysis modules, user interfaces, and decision support systems.
- Clinical workflow design: Integrate clinical processes such as symptom recording, automatic diagnosis, and recommendation provision.
- Select technology: Select a development platform, database, and AI/ML technology for predictive analysis.
- Design a system model, including data flow, decision process, module integration, process flow diagrams, and system architecture.
- Design a user interface based on clinical user needs.
- Select technology such as databases, cloud platforms, AI algorithms, or analytical frameworks.
- Results: System architecture diagram and initial prototype.

The following is a detailed explanation of the System Architecture Diagram and Initial System Prototype of the Clinical Group Decision Support System (CGDSS) for dengue management as follows:

Table 2. System Architecture Diagram

This diagram reflects the system structure from the data source level to the user interface.	
1. Data Sources Layer:	- Electronic Medical Record (EMR): Contains patient data such as medical history, diagnosis, and treatment. - Laboratory Results: Information such as NS1 antigen, platelet, and hematocrit test results. - Epidemiological Data: Patient geolocation data and case cluster reports from health authorities. - Environmental Data: Weather, humidity, and temperature information relevant to dengue spread patterns.
2. Integration Layer :	- API and Middleware: Connect data sources with the CGDSS core system. - Data Processing Engine: Performs data extraction, transformation, and loading (ETL) for format standardization. - Data Security: Uses encryption protocols to protect patient data.
3. Application Layer :	- Decision Algorithm Module: Uses rule-based algorithms and AI for predictive analysis and clinical recommendations. - Epidemiological Analysis Module: Identifying case spread patterns and cluster risks. - Report Management: Creates case reports, outbreak trends, and data visualization.
4. Presentation Layer:	- Clinical Dashboard: Provides diagnosis and treatment recommendations to doctors. - Epidemiology Map: Geographical visualization of dengue outbreak clusters. - Notification: Sends spike alerts to health workers. - Report: Weekly and monthly case reports in downloadable formats.
5. Pengguna Sistem:	- Doctors: Need clinical recommendations. - Nurses: Use for patient monitoring. - Public Health Managers: Monitor trends and create policies.

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

The following is a detailed explanation of the Initial Prototype of the Clinical Group Decision Support System (CGDSS) for dengue management as follows:

Table 3. Initial Prototype of the System

The initial prototype describes a basic user interface (UI) for CGDSS. Description of each component:	
1. Main Dashboard:	- Displays patient data summary, current dengue case statistics, and trend graphs. - Simple navigation with tabs for key features: Diagnosis, Monitoring, and Reports.
2. Diagnosis Module:	- Symptom and Patient Data Input: Form to enter symptoms, laboratory results, and additional information. - Recommendation Results: Displays recommended diagnosis and treatment based on WHO protocols.
3. Outbreak Monitoring Map:	- Interactive map with markers of active cases, high-risk clusters, and areas of concern. - Data Filter: Based on time, location, and severity.
4. Report Module:	- Page to create epidemiological reports, trends, and spread predictions. - Export options to PDF, Excel, or share via email.
5. Notification:	- Pop-up panel to provide spike case alerts or high-risk alerts.
Interative Development	The initial prototype only includes core features. After user validation (doctors, nurses, and health managers), the system can be further developed with: - Machine Learning Integration: To improve the accuracy of predictive analysis. - Chatbot: Provide recommendations or quick answers for medical personnel. - Mobile Access: Supports users on mobile devices.

3. Development and Implementation

At this stage, the system is developed based on the design that has been prepared. The main activities are:

- Coding: Building software modules to support CGDSS functions such as patient data management, trend analysis, and report generation.
- System integration: Connect CGDSS with electronic medical record (EMR) and laboratory systems to obtain data.
- Initial testing: Unit tests are conducted on each component to ensure essential performance.
- Result: Initial version of the CGDSS system.

Implementing a Clinical Group Decision Support System (DCGDSS) for Dengue Management is integrating advanced technology into healthcare to assist medical professionals in diagnosing, planning care, and making clinical decisions related to patients infected with dengue fever. The following is an explanation of the main steps in its implementation:

Table 4. Development and Implementation Stages

1. Data Collection	- Data Sources: Collect data from multiple sources, such as electronic medical records (EMRs), lab results, clinical reports, and environmental data, such as weather patterns that affect mosquito distribution. - Real-Time Data: IoT (Internet of Things) sensors and wearable devices can help collect real-time patient data, such as body temperature, blood pressure, and fluid levels.
2. Predictive Model Development	- Machine Learning (ML): Machine learning algorithms

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

	analyze collected data and generate predictions about disease progression, severity, and treatment recommendations. - AI Integration: AI-based systems can provide personalized treatment recommendations based on patient profiles.
3. Collaborative Systems Design	- Information Sharing Platform: Create a cloud-based system that allows doctors, nurses, and other healthcare professionals to share patient information and analysis results quickly and securely. - Multidisciplinary Discussion: The system supports virtual discussion forums to integrate perspectives from multiple medical teams (e.g., infectious disease specialists, epidemiologists, and nutritionists).
4. Key Features in the System	- Decision Tools: Provide evidence-based guidelines for diagnosing dengue, determining the need for hospitalization, and monitoring complications such as dengue shock syndrome. - Early Warning**:** The system can notify the medical team about potential complications that require rapid intervention. - Data Visualization: Interactive dashboards to display patient data, epidemic trends, and severity levels in an easy-to-understand manner.
5. Training and Adaptation	- Medical Team Training: Provide system users with training to ensure a good understanding of the system's functions and operations. - Feedback Loop: Integrate feedback mechanisms to improve the system based on field experience.
6. System Evaluation and Improvement	- Effectiveness Analysis: Evaluate the system's effectiveness in reducing mortality and morbidity and improving hospital operational efficiency. - Data-Driven Adjustment: The system must be continuously updated based on the latest medical developments and dengue epidemic trends.
Key Benefits	- Fast and Accurate Decision Making: The system helps the medical team make decisions based on data and evidence-based recommendations. - Enhanced Collaboration: Allows different specialties to work together on a single platform. - Complication Prevention: The system detects the risk of complications early, allowing proactive action.

4. Validation

The validation phase ensures the system meets the requirements and functions well in real-world scenarios. The steps include:

- Clinical trials: Using the system in a limited clinical setting to identify bugs or weaknesses.
- Validation with accurate data: Historical dengue data is used to test the accuracy and relevance of the system's decisions.
- User evaluation: Collecting user feedback to improve the interface and workflow.
- Output: A tested and refined version of the system.

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

Validation of the Developing Clinical Group Decision Support System (DCGDSS) for Dengue Management is crucial to ensure the system can function adequately, provide accurate recommendations, and be accepted by users (medical teams). The following are the common validation stages:

Table 5. CGDSS Validation for Dengue Management

1. Conceptual Validation	- Objective: Ensure that the system design meets clinical needs in dengue management.
	- Steps:
	  - User requirements are reviewed by experts (specialist doctors, nurses, epidemiologists, etc.).
	  - Discussion with stakeholders to evaluate whether the system workflow reflects actual clinical practice.
	  - Simulation of the system workflow to identify deficiencies at an early stage.

2. Data Validation	- Objective: Ensure that data entered, processed, and generated by the system is accurate and consistent.
	- Steps:
	  - Verify data integrity from electronic medical records (EMR) or laboratory results.
	  - Test the quality of data processing algorithms to ensure that input errors do not affect the results.
	  - Validate machine learning algorithms with labeled datasets to ensure accurate predictions.

3. Functional Validation	- Objective: Test whether all system features function according to specifications.
	- Steps:
	  - Trial all modules, such as patient data input, clinical analysis, and delivery of treatment recommendations.
	  - Simulation of clinical scenarios to ensure that the system provides logical recommendations that are in accordance with medical guidelines.
	  - Usability testing of the dashboard by users to ensure accessibility and ease of use.

4. Clinical Validation	- Objective: Ensure the system's recommendations are consistent with current clinical guidelines and valuable to the medical team.
	- Steps:
	  - Test the system on real dengue cases compared to decisions made by physicians.
	  - Focus group discussions with the medical team to evaluate the relevance and usefulness of the system's results.
	  - Comparison of the system's decision results with manual results or standard practices.

5. User Validation	- Objective: Ensure that the system is comfortable to use and accepted by end users.
	- Steps:
	  - Conduct training for users and gather feedback on their experiences.
	  - Trial the system in a real clinical setting with the involvement of physicians, nurses, and other staff.
	  - Survey or interview to understand user satisfaction levels and areas for improvement.

6. Security Validation	- Objective: Ensure the system is secure from data leaks or cyber threats.

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

	- Steps: - Testing patient data encryption to protect confidentiality. - Simulating cyber attacks to ensure the system is resilient to external threats. - Compliance with legal and regulatory standards, such as HIPAA or local health data regulations.
7. Performance Validation	- Objective: Assess the speed and stability of the system under various conditions. - Steps: - Load testing to see how the system handles multiple users or data simultaneously. - Measuring the system's response time, especially for data analysis and recommendation features. - Testing under extreme conditions, such as slow networks or large data volumes.
8. Field Testing	- Objective: Test the system thoroughly in a real operational environment. - Steps: - Implementation of the system in several health facilities as a pilot project. - Direct monitoring of system performance, including input from medical staff. - Document emerging issues and implement needed improvements.
9. Final Validation	- Objective: Conclude that the system is ready for widespread use. - Steps: - Compile an evaluation report that includes all previous validation results. - Obtain approval from the supervisory team, regulator, or related medical institution. - Final adjustments before the official launch of the system.

5. Deployment

This stage is the full rollout of the system in an operational environment. The main activities are:

- User training: This involves training doctors, nurses, and technical staff on how to use the system.
- Full-scale deployment: The system will be implemented in all targeted facilities, such as hospitals, clinics, and community health centers.
- Monitoring and maintenance: Monitoring system performance, updating features as needed, and addressing technical issues.
- Result: A fully functional CGDSS system to support decision-making in dengue management.

The stages of the Developing Clinical Group Decision Support System (DCGDSS) system deployment for dengue management involve several systematic steps to ensure the system runs well and provides optimal benefits. The following are the details of the stages:

Table. 6 Stages of the Developing DCGDSS system Deployment for dengue management

1. Deployment Planning	- Identification of infrastructure needs: Identifying the hardware (servers, computers, networks) and software needed. - Determination of deployment scope: This involves determining which units or locations, such as hospitals, clinics, or health centers, will use the system. - Timeline preparation: Making a gradual deployment implementation time plan.
2. Infrastructure Preparation	- Hardware installation: Ensure servers, networks, and related

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

	devices are ready. - Network configuration: Set up the network to support communication between devices connected to the DCGDSS. - Security settings: Implement firewalls, encryption, and authentication systems to protect patient and system data.
3. System Installation	- Software installation: Install DCGDSS on servers and devices used by health workers. - Initial system testing: Conduct testing to ensure the installation succeeds and all components can work. - Integration with other systems: If necessary, integrate DCGDSS with the hospital information system (SIR) or laboratory.
4. Settings and Configuration	- System parameterization: Configure the system according to needs, such as location, number of users, and data format. - User access creation: Determine access rights and profiles for health workers based on their roles. - Initial data input: Enter initial data such as patient data and dengue case history.
5. User Training	- Technical training for the IT team: Teach how to handle technical issues that may arise. - Operational training for health workers: Guide on using DCGDSS in dengue case management. - Usage simulation: Conduct trials using fictitious cases to ensure users understand the system.
6. Final Testing (Validation)	- End-to-end trials: Test the system with actual workflows to ensure system performance is as expected. - Stress testing: Test the system's ability to handle data volumes and users in heavy scenarios. - Bug identification: Fix issues or bugs found during testing.
7. Implementation (Go-Live)	- System rollout: Start using the system to support clinical decisions in dengue management. - Initial monitoring: The IT team monitors the system for the first few weeks to ensure stability. - Feedback collection: Get input from users for further improvements.
8. Maintenance and Evaluation	- System updates: Install software updates to improve functionality or security. - Performance monitoring: Ensure the system runs smoothly through regular monitoring. - Success evaluation: Assess the impact of DCGDSS use on dengue case management, including its efficiency and effectiveness.

6. Maintenance

- Objective: Maintain system performance and sustainability.
- Steps:
  - Monitor system performance periodically.
  - Fix bugs and update features based on user feedback.
  - Ensure patient data is secure according to regulations (e.g., HIPAA or GDPR).
  - Make updates to support evolving clinical needs.
- Outcome:
  - A reliable, secure system that continues to support clinical decisions effectively.

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

Maintenance of Developing a Clinical Group Decision Support System (DCGDSS) system for dengue management includes a series of stages to ensure the system runs smoothly, accurately, and reliably. The following are the maintenance stages in detail:

Table 7. Stages of maintenance of CGDSS system for dengue management

1. Maintenance Needs Analysis	- Objective: Identify maintenance needs based on system performance, user reports, and analytical data. - Activities: - Collect feedback from system users (doctors, medical personnel, IT team). - Evaluate system performance data (response time, prediction accuracy, operational stability). - Identify technical or functional issues.
2. Preventive Maintenance	- Objective: Prevent system disruptions before they occur. - Activities: - Perform routine software updates, such as patching for bugs and security - Ensure the system database is always organized and free from data duplication. - Optimize analytical algorithms to maintain prediction accuracy. - Adjust the system to the latest health protocols related to Dengue.
3. Corrective Maintenance	- Objective: Fix identified problems. - Activities: - Analyze error logs to find the source of the problem. - Fix modules that do not function according to specifications. - Recover lost or corrupted data using backups. - Overcome system integration constraints with other tools or platforms (such as medical devices or epidemiological data).
4. System Updates and Development	- Objective: Improve system features and capabilities based on evolving needs. - Activities: - Add new features such as more intuitive data visualization or more sophisticated predictive analytics. - Improve compatibility with new hardware or other systems. - Test new modules in a simulated environment before implementation.
5. Continuous Monitoring and Testing	- Objective: Maintain optimal system performance. - Activities: - Monitor system parameters such as real-time response rate, error rate, and server load. - Automated testing tools are used to check module integrity. - Conduct stress testing to ensure the system can handle spikes in usage.
6. Documentation and Training	- Objective: Ensure that all system changes are well documented and that users understand the updates. - Activities: - Create technical documentation about software changes or system configurations. - Provide training to users on new features or changes to the interface. - Provide troubleshooting guides for users.
7. Evaluation and Feedback	- Objective: Assess maintenance effectiveness and identify areas

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

for improvement. - Activities: - Conduct user satisfaction surveys. - Review system performance reports over some time. - Develop long-term maintenance plans based on the evaluation results.

CONCLUSION

The development and implementation of CGDSS for Dengue management with System Engineering Process Methodology is more structured, flexible, and adaptive and can overcome various problems related to data integration, algorithm accuracy, and user acceptance. The stages consist of Requirements Analysis, System Design, Implementation, Testing, Deployment, and Maintenance. System Engineering Process Methodology can be applied to Developing Clinical Group Decision Support System for dengue management. In addition, the research results can also be used for different cases such as infectious diseases, namely Influenza, COVID-19, chickenpox, hepatitis B, HIV / AIDS, Tuberculosis (TB), pneumonia, diphtheria, leprosy, Malaria, toxoplasmosis, Candidiasis, dermatophytosis, and malaria. Non-communicable diseases, namely Diabetes mellitus, hypertension, coronary heart disease, Osteoporosis, osteoarthritis, Depression, schizophrenia, Cancer, and lungs.

REFERENCES

[1]. Darbi, H. A., & Saleh, E. M. (2022). Decision support system: Analysis and design methodology. *2022 IEEE International Conference on Artificial Intelligence, Modeling, and Simulation Technologies and Applications (MI-STA)*. https://doi.org/10.1109/MI-STA54861.2022.9837769

[2]. Henshaw, M., & MacDonald, A. (2022). Systematic engineering processes for development of complex DSS. *Journal of Systems Engineering and Electronics*, *35*(4), 335–348.

[3]. Jackson, J., & Lin, M. (2023). Hybrid engineering approaches for DSS: A systems engineering perspective. *Journal of Engineering Systems Design*, *12*(1), 21–33.

[4]. Kawamoto, K., Houlihan, C. A., Balas, E. A., & Lobach, D. F. (2021). Clinical decision support systems for the practice of evidence-based medicine: A systematic review. *Journal of the American Medical Informatics Association*, *28*(3), 567–573.

[5]. Mora, M., Wang, F., Gelman, O., & Kljajic, M. (2011). IDSSE-M: A software system engineering methodology for developing intelligent decision-making support systems. *International Journal of Decision Support System Technology*. https://doi.org/10.4018/jdsst.2011100104

[6]. Martin, J., & Lane, R. (2021). Integrating agile and waterfall methods in system engineering for DSS development. *International Journal of Computer Applications in Technology*, *65*(2), 178–190.

[7]. Melin, J., & Papalambros, P. (2022). Designing group decision support systems for collaborative decision-making in complex engineering projects. *International Journal of Decision Support System Technology*, *14*(2), 58–74.

[8]. Sage, A. P., & Rouse, W. B. (2020). *Handbook of systems engineering and management* (4th ed.). Wiley.

[9]. Smith, P., & Lee, H. (2020). Adapting system engineering methodologies for real-world DSS applications. *Journal of Decision Systems*, *29*(6), 500–514.

[10]. Vallance, S., Duffy, A. H. B., Whitfield, R. I., Mendibil, K., Hird, A., McCabe, J., & Turner, N. (2011). Using systems engineering for the development of decision-making support systems (DMSS): An analysis of system development methodologies (SDM). *ResearchGate*. https://doi.org/10.4018/978-1-4666-4002-3.ch002

[11]. Lawrence, S., & Herber, D. R. (2024). A model-based systems engineering approach for effective decision support of modern energy systems depicted with clean hydrogen production. *Systems*, *12*(8), Article 290. https://doi.org/10.3390/systems12080290

[12]. Turban, E., Sharda, R., & Delen, D. (2021). *Business intelligence and analytics: Systems for decision support* (11th ed.). Pearson.

[13]. Verma, P., & Gupta, S. (2021). Applying system engineering processes to decision support systems development. *International Journal of Engineering and Technology Innovation*, *10*(3), 209–220.

[14]. Zhang, Y., Liu, H., & Lu, J. (2021). Clinical group decision support system based on artificial intelligence for multi-disciplinary medical teams. *Artificial Intelligence in Medicine*, *117*, 102069.