

A Comprehensive Decision Support System for Family Physicians

Emre Şimşek, Fatma Ezgi Erkat, İlker Korkmaz, Ahmet Sermet Anagün

Faculty of Engineering

Izmir University of Economics

Izmir, Türkiye

{emre.simsek, ezgi.erkat}@std.ieu.edu.tr, {ilker.korkmaz, sermet.anagun}@ieu.edu.tr

Abstract—Family physicians practice in demanding clinical environments where quick yet reliable decision-making directly affects patient outcomes. This paper introduces the design and evaluation of a lightweight, offline-capable Clinical Decision Support System (CDSS) developed specifically for the realities of family medicine. Built with open-source tools such as Python, PyQt5, and SQLite, the system combines a transparent decision-tree engine, a simple graphical interface (GUI), and an administrative dashboard for monitoring key performance indicators (KPIs). The development process is described step by step, from architectural design and data handling to interface design and usability testing. Simulated patient cases suggest that the system improves diagnostic efficiency and consistency while maintaining user satisfaction.

Keywords—clinical decision support system, family medicine, decision tree, key performance indicator.

I. INTRODUCTION

The healthcare system increasingly depends on front-line practitioners making accurate decisions under the limited time and resource constraints [1]. Family physicians, who are often the first point of contact, face a particularly heavy cognitive workload: they should manage diverse symptoms, treat chronic conditions, and handle preventive care within very limited consultation windows [2]. This challenge is compounded by fragmented patient records and administrative responsibilities [3, 4].

In some cases, family physicians may be the only accessible healthcare providers. Possible errors in diagnosis processes can have serious consequences, yet an advanced diagnostic infrastructure is often unavailable [1]. Therefore, low-cost and adaptable decision support systems that are also capable of operating offline are needed [5].

Clinical Decision Support Systems (CDSSs) have shown potential to reduce diagnostic variability and improve efficiency [1, 5]. However, many systems fail to gain traction in primary care because they present overly complex interfaces, hide decision logic, or create a perception of over-reliance on automated results [3]. Physicians tend to prefer tools that support rather than replace clinical judgment, offering guidance while preserving autonomy [4]. The proposed CDSS presented in this paper was designed with this philosophy in mind. It is offline-capable, modular, and open-source. Decision

logic is stored in human-readable JavaScript Object Notation (JSON) files, making it easy to update without programming knowledge [3]. The graphical user interface (GUI) is intentionally simple to guide physicians step by step through a decision tree; meanwhile the interactions are logged for later analysis [6, 7]. An integrated analytics module aggregates user data into key performance indicators (KPIs). Such features aim to provide the system to be used as a practical and scalable tool for family physicians.

The rest of the paper is organized in the following structure: Section II gives the related work about the initial times of CDSS research, Section III explains the design and implementation of the proposed CDSS, Section IV summarizes the preliminary test results based on the simulated patient data, and finally Section V concludes the paper with mentioning possible future work.

II. RELATED WORK

The concept of computer-assisted diagnosis has evolved for more than five decades. Early systems such as MYCIN [8] and INTERNIST [9] demonstrated that structured medical knowledge could be represented computationally using rule-based reasoning. However, their static rule sets limited the broader use of these computer programs in adaptability-required real world scenarios [10].

In the 1990s and early 2000s, CDSS research shifted toward probabilistic models, enabling systems to better handle uncertainty [5]. Systems like Isabel Healthcare [11] expanded diagnostic coverage and were integrated with electronic health records (EHRs), but such integration was often provided with increased complexity and heavy IT support [1].

More recently, advances in machine learning and natural language processing have powered modern CDSSs platforms [1]. These systems can help professionals classify the diagnosis using pattern recognition technologies. Nevertheless, if such a system offers a “black-box” model on the decision paths, the system would lack transparency and so the physician’s trust on the system would be decreased [3, 4].

Comparative studies suggest that structured approaches, such as decision trees and flowcharts, are more acceptable to clinicians because they clearly display the reasoning process [1, 4]. Visual representations of diagnostic pathways allow

physicians to validate system outputs that strengthen trust and adoption [6]. In addition, usability studies stress that a CDSS should minimize data entry and align with clinical workflows [7].

The proposed system draws from this tradition, combining decision-tree logic with the adaptability of transparent (i.e., human-readable) JSON-based structures, all implemented through accessible open source tools.

III. SYSTEM DESIGN AND IMPLEMENTATION

The architecture of the proposed CDSS was deliberately designed to be modular, lightweight, and adaptable, ensuring that each component could evolve without disrupting the system as a whole. The solution integrates the following components: Decision Engine, GUI, and Administrative Dashboard, all connected to a centralized data layer.

A. System Architecture

The high-level architecture depicted in Fig. 1 illustrates the interaction between the Decision Engine, GUI and Data layers. The Decision Engine interprets JSON-encoded decision trees and navigates the physician step by step through a structured diagnostic process. The GUI serves as the primary interaction point, while the SQLite-based data layer manages patient records, logs, and complete Q&A trails.

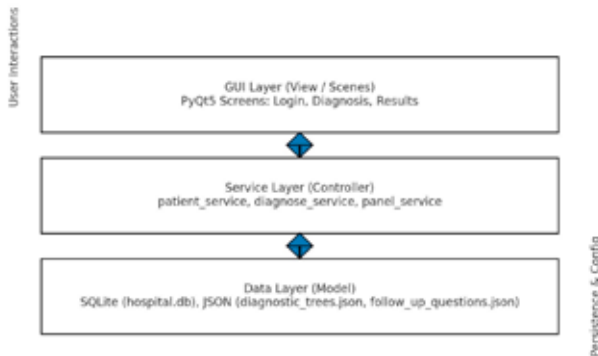


Fig. 1: The system architecture.

This modular separation ensures flexibility. For example, the decision tree model could be extended with probabilistic weights without altering the GUI, or the SQLite backend could be migrated to a client-server database if multi-user deployment is required.

B. Data Flow

The workflow begins with user authentication and patient record retrieval. Once a case is initiated, the Decision Engine starts from the root node of the chosen decision tree. Each question, corresponding responses, and transition are logged in real time, preserving the diagnostic pathway. Once a terminal diagnosis is reached, the result is stored in the database and made available within the administrative dashboard. The structure shown in Fig. 2 represents a sample decision tree for a migraine symptom encoded in the system.



Fig. 2: A sample decision tree for a migraine symptom.

This structured approach ensures that performance monitoring remains independent from patient care while still enabling a comprehensive set of analytical capabilities.

C. Decision Logic Implementation

Decision logic is structured as a binary tree and is encoded in JSON format, with each node representing a question, possible responses, and links to subsequent nodes or terminal outcomes. This approach balances human readability with computational efficiency, enabling clinicians or administrators without programming expertise to update diagnostic protocols.

The traversal of the tree is implemented in the following manner: the current node is displayed with its question, the physician's response is recorded, and the system transitions to the next node or outcome. All interactions along the path from the first node to the last inclusively are stored, creating an auditable log for both quality assurance and medical legal requirements.

D. Graphical User Interface

The GUI, implemented using PyQt5, was designed for clarity and ease of use. In deed, the essential functions of the system were focused to meet the basic requirements of the family physicians. The GUI comprises three primary screens: login, diagnosis, and results.

The login screen shown in Fig. 3 allows physicians authenticate themselves and select or create a patient record. This ensures that diagnostic sessions are linked to the appropriate user and patient.



Fig. 3: Login screen for a family physician.

The diagnosis screen shown in Fig. 4 displays one question at a time with the corresponding response options. Such a sequential presentation helps clinicians maintain focus and reduces cognitive burden, as recommended by prior usability research [6, 7].



Fig. 4: Diagnosis screen displaying active options.

The results screen shown in Fig. 5 provides the family physicians with the final diagnosis, ICD-10 codes, and a complete trace of the reasoning process. Physicians can review the diagnostic path on the decision tree before final confirmation, reinforcing trust and transparency.

E. Administrative Dashboard and KPI Analytics

The Administrative Dashboard aggregates usage data and transforms it into actionable insights. The KPIs like average diagnosis time, physician-specific performance scores, revisit rates, and diagnostic frequency distributions are automatically computed and visualized.

The histogram in Fig. 6 allows administrators to compare individual physician scores with clinic-wide averages with the aim of identifying both high performers and those requiring additional training.



Fig. 5: Results screen summarizing the session history.



Fig. 6: Doctor scores against clinic-wide average.

The heatmap in Fig. 7 visualizes performance variation across time and individual physicians, aiding in resource planning and quality improvement efforts. Beyond this, the heatmap can highlight seasonal workload effects (e.g., influenza season) on physician performance, making it easier to anticipate periods that may require additional staffing or resources. It also exposes instances where specific physicians show decreasing performance over time, enabling targeted support or training interventions. In this way, the dashboard serves not only as a retrospective reporting tool but also as a proactive management resource.

F. Database Schema

The SQLite database schema of the system includes two primary tables: patients and diagnoses. The patients table stores demographic and identifying information, while the diagnoses table records session-specific data, including the Q&A trail and final outcome. This separation ensures clinical data integrity while supporting performance monitoring.

IV. RESULTS AND EVALUATION

Experiments were conducted using a combination of simulated patient cases and direct usability tests. The simulations covered both common and less frequent scenarios to assess the robustness of decision logic. A family physician involved in the development and evaluation phases of the system reported that the preliminary findings of the system has a potential to

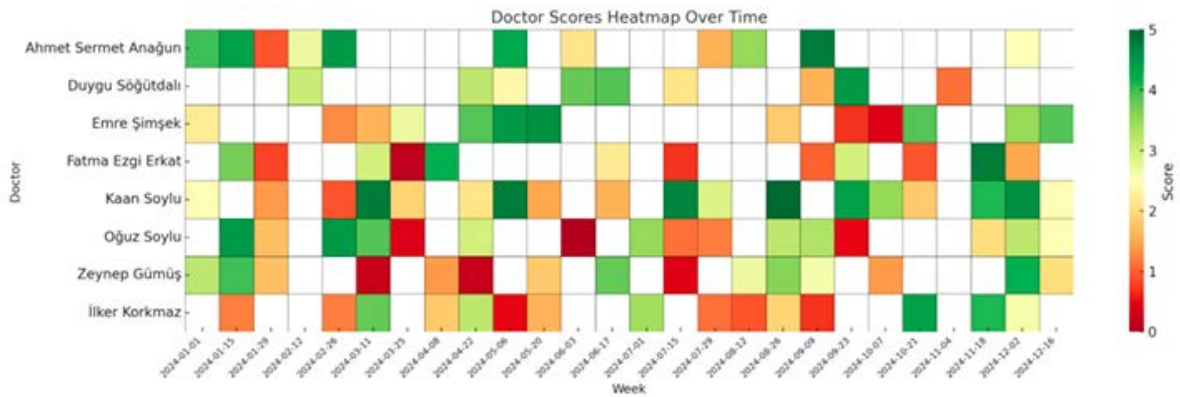


Fig. 7: Heatmap of weekly performance scores across doctors.

reduce cognitive workload and the design of the system has a prospect to increase clarity in diagnostic reasoning.

Quantitatively, the average diagnostic session was completed in less than 2.5 minutes, a result consistent with prior findings on workflow-friendly CDSSs [1, 2]. Qualitative feedback emphasized the transparency of decision trees and the usefulness of reviewing the reasoning process before finalizing an assessment [3, 4].

Nevertheless, these evaluations were conducted under controlled conditions. The simulation-based patient data setting and the number of physicians who tested the system are some of the main limitations for now. The definite efficiency of the system in busy clinical environments, where interruptions and patient variability are significant, remains to be validated.

V. CONCLUSION AND FUTURE WORK

This study introduced and evaluated a modular CDSS proposed for the use of family physicians. By integrating JSON-based decision trees, a lightweight PyQt5 GUI, and KPI-oriented analytics, the proposed system addresses both clinical and administrative needs.

The system's transparency and adaptability distinguish it from many commercial platforms. Physicians can examine and update decision logic themselves, which may help overcome the trust barrier that often hinders CDSS adoption [3, 4]. The system's significant characteristics, (i.e., physician oriented sequential question design, clean interface, and offline capability) align with the realities of everyday practice. The preliminary results indicate that the system has a potential to reduce session times and cognitive load while standardizing assessments.

From an administrative perspective, the analytics module linked to the administrator dashboard supports data-driven performance management. By tracking selected KPIs such as consultation times, diagnosis frequencies, and physician-specific metrics, administrators can plan targeted interventions.

Still, important limitations remain. The system has not yet been tested in real-world clinics, where integration with existing EHRs and patient diversity could present challenges. Hence, future work should explore broader field validation

across varied practice environments. Other possible future work would include (i) integration with national health information exchanges to reduce duplicate data entry, (ii) plugging the natural language processing technologies for free-text symptoms, and (iii) incorporating automated alerts for abnormal KPI trends.

In summary, the proposed CDSS shows promise as a practical and adaptable decision support tool for family medicine. The potential of the system, however, would be clear after extended deployment and evaluation in real clinical practice.

REFERENCES

- [1] R.T. Sutton, D. Pincock, D.C. Baumgart, D.C. Sadowski, R.N. Fedorak, and K.I. Kroeker, "An overview of clinical decision support systems: benefits, risks, and strategies for success," *npj Digital Medicine* vol. 3, no. 17, 2020, doi: 10.1038/s41746-020-0221-y.
- [2] H.J. Michtalik, H.C. Yeh, P.J. Pronovost, and D.J. Brotman, "Impact of attending physician workload on patient care: a survey of hospitalists," *JAMA Internal Medicine*, vol. 173(5), pp. 375-377, 2013, doi: 10.1001/jamainternmed.2013.1864.
- [3] E.S. Berner, *Clinical Decision Support Systems: Theory and Practice*, 2nd edition, Springer, 2007, doi: 10.1007/978-0-387-38319-4.
- [4] D. Zikos and N. DeLellis, "CDSS-RM: a clinical decision support system reference model," *BMC Medical Research Methodology*, vol. 18, no. 137, 2018, doi: 10.1186/s12874-018-0587-6.
- [5] J.A. Osheroff, J.M. Teich, D. Levick, L. Saldana, F.T. Velasco, D.F. Sitig, K.M. Rogers, and R.A. Jenders, *Improving Outcomes with Clinical Decision Support: An Implementer's Guide*, 2nd edition, HIMSS, 2012, doi: 10.4324/9781498757461.
- [6] B.T. Karsh, M.B. Weinger, P.A. Abbott, and R.L. Wears, "Health information technology: fallacies and sober realities," *Journal of the American Medical Informatics Association*, vol. 17(6), pp. 617-623, 2010, doi: 10.1136/jamia.2010.005637.
- [7] D. Armijo, C. McDonnell, and K. Werner, "Electronic Health Record Usability: Evaluation and Use Case Framework," Agency for Healthcare Research and Quality, publication no. 09(10)-0091-1-EF, 2009.
- [8] E.H. Shortliffe, "A Rule-Based Computer Program for Advising Physicians Regarding Antimicrobial Therapy Selection," *Proceedings of the 1974 Annual ACM Conference - vol. 2*, p. 739, 1974, doi: 10.1145/1408800.1408906.
- [9] H.E. Pople, "The Formation of Composite Hypotheses in Diagnostic Problem Solving: An Exercise in Synthetic Reasoning," *Proceedings of the Fifth International Joint Conference on Artificial Intelligence (IJ)*, pp. 1030-1037, 1977.
- [10] M.A. Musen, Y. Shahar, and E.H. Shortliffe, "Clinical Decision Support Systems," in *Biomedical Informatics: Computer Applications in Health Care and Biomedicine*, 3rd edition, E.H. Shortliffe and J.J. Cimino, Eds., Springer, 2006, pp. 698-736, doi: 10.1007/0-387-36278-9_20.
- [11] Isabel Healthcare Ltd., <https://www.isabelhealthcare.com>, (accessed on 11 July 2025).