Development of a Scalable Tele-Monitoring Platform based on the Clinical Decision Support Systems

azita yazdani (a_azdani@sums.ac.ir)
Shiraz University of Medical Sciences https://orcid.org/0000-0002-5190-286X
Reza Safdari
Tehran University of Medical Sciences
Roxana Sharifian
Shiraz University of Medical Sciences
maryam zahmatkeshan
Fasa University of Medical Science

Software

Keywords: Service Oriented Architecture, Clinical Decision Support System, Telehealth, Scalability, Systems integration, Health Information Interoperability, Electronic Medical Record

Posted Date: August 18th, 2020

DOI: https://doi.org/10.21203/rs.3.rs-49540/v1

Abstract

Background: One of the most important types of information systems that play important role today in providing quality health care services are clinical decision support systems (CDSSs). These systems are effective in overcoming human resource constraint and intelligent analysis of information generated by Tele-monitoring systems. In spite of the many advantages of this architectures, these are single-purpose, meaning that only the CDSS of a disease is located on them. If we want to use the same model of architecture in the decision-making process of another disease, all the components of this architecture should be redevelopment with a new CDSS, which is time-consuming and costly. Due to the increasing demand for health information technology at low cost and mobile access in the health care industry, in this article, a scalable software platform(Patient Tele monitoring: PATEL) based on SOA for implementing and use different CDSSs on a common platform, for use in Tele-monitoring Systems, was created.

Implementation: To develop PATEL platform, the component-based software development approach and hybrid programming approach to implementing various components used. In the evaluation phase of the proposed platform, the case study, accuracy and performance evaluation (transmission delays, patient data fetch, parsing overhead and inference time) used.

Results: The results of the case study evaluation confirmed the scalability and interoperability between CDSSs on the platform. Based on performance evaluation, the proposed platform has responded to 89% of the requests in less than one second. Also, based on accuracy evaluation, the platform presented in this article was successful in diagnosing 91.6% of the cases.

Conclusion: The proposed platform can support CDSSs of various diseases simultaneously and provides the necessary scalability to add a new CDSS. Tele-monitoring systems will be capable of service by connecting to this platform. Using this infrastructure is expected to be a lot of duplication in the implementation of tele-monitoring systems based CDSSs will be reduced.

Background:

Today, decision support systems are used to improve the decision-making process at different decision-making levels[1]. These systems are either standalone or integrate with other systems[2]. In order to create decision support systems, as in any other system, system development steps must be taken into account: including identification of requirements, design, development, testing, and ultimately implementation. Each of the phases of the system development has its own time and cost[3]. When CDSSs are created, using solutions to make use of these systems on a large scale can reduce the cost of creating and maintaining these systems and with their subscription, will prevent the production of multiple and island systems[4, 5]. In recent years, one of the approaches used in combination with CDSSs for this purpose is the service-oriented architecture (SOA) approach [6-9]. In SOA as an integration strategy[10], decision support systems are available as a service to the client (individuals or systems) and provide benefits such as facilitating knowledge storage, reducing costs, and improving agility[6]. CDSS based Tele-monitoring systems and decision support telemedicine systems [11] are an emerging technology that is rapidly developing that benefit from the advantages of SOA. The explosion of information and the need for communication between health care providers are the starting point for such systems[4]. But this type of health care system faces many challenges, including certain use case and low-resilience to change[4, 12]. However, most of the existing CDSSs only for one type of disease with a single knowledge base, limited decision-support content is difficult to meet the needs of clinicians, hindering the application of the CDSSs. So that we need to make a lot of changes if we want to modify a CDSS-based Tele-monitoring system designed for a disease that can be used for another disease. These changes will be very large, meaning that many components of this system should be created from the beginning. The problem of non-flexibility and scalability in these systems has led many Tele-monitoring systems to be developing, while many of these systems have a common structure. so they have island system barriers[13]. In this paper, in order to solve the problem of the flexibility and scalability of these systems, a platform proposed that allows CDSSs to be placed on a common Infrastructure and the interaction between CDSSs is provided which increases the synergy between them. This platform has integration with Electronic Health Record (EHR) and can provide high flexibility to simultaneously support knowledge base of various diseases. In this platform, flexibility is provided by creating a number of specific modules that are available in most CDSS-based Tele-monitoring systems. By using these modules, and adding knowledge base of various disease, the ability of Tele-monitoring of multiple diseases based on CDSSs on a common platform is provided. Utilizing the proposed model in this paper will reduce the set-up costs and speeding up the launch of various CDSS-based Tele-monitoring systems, and many duplications in the implementation of these systems.

Implementation:

The present study was conducted in four phases: 1: determining platform modules, 2: Conceptual model design (We have published the details of these two steps in another article[4]), 3**:**developing platform and finally evaluating. To development of PATEL platform, component-based software development approach used and since different components, have different structure and approach in the implementation, so different techniques and languages used to implement them. In the following, all software used to implement the platform introduce. Access for users (health care team and patient) implemented into the platform user interface with Google Android Studio. Platform Management Panel to define the health care team, diseases, symptoms, parameters, and CDSSs have been launched. This section has been developed using PHP programming language. The database of this platform implemented with MySQL and Web services with JSON. The inference engine of CDSSs in the platform is CLIPS and uses the "PHP-CLIPS" plugin as the interface for communicating between PHP and CLIPS.

a. **Platform architecture design**: The conceptual model of our proposed platform as showed in Figure 1. The system admin interface, which is responsible for defining the doctor, disease, symptoms and adding clinical decision support systems for various diseases, is implemented with PHP. The health care team are the second group of users of this platform that can by Android-based interface define a new patient for monitoring, access to patient data (EHR), define a tele-monitoring program for patients and reporting health status of patients. Patient measure and transmit the symptoms associated with his disease by the Android-based interface. These symptoms identified by the health care team for the monitor at the specified periods. The platform sends patient data for CDSSs in real-time. The inference results as the patient's conditions (Normal, Warning, and Critical) and recommendations sent to the health care team and the patient. This platform has three servers. The first server is the management of requests, which specifies whether a request is to inference patient conditions or to access the EHR. The second server is the EHR Server, where the patient's data and history recorded. On the third server, there are clinical decision support systems.

b. **Adding/editing clinical decision support systems:** The most important feature of the proposed platform that distinguishes it from other existing architectures is its scalability. This feature means the ability to add new CDSSs to the platform. Due to difference between varied diseases, the medical knowledge, clinic rules used to treatment different patients are not same either.so our platform's scalability feature makes it suitable for intelligent monitoring of diverse diseases on a common infrastructure. As it mentioned, in this platform the common components of tele-monitoring systems based on CDSSs are provided. Due to the provision of components of decision support systems, it is only necessary to have a knowledge base for monitoring various diseases, for which purpose it is possible to upload pre-created knowledge bases or using the management panel to create a new knowledge base. Using the multi-threaded programming technique, the platform is used in parallel for different CDSSs at inference time. The admin can upload the CDSSs as a CLIPS file (These knowledge bases have already been developed) or configuring the clinical knowledge by the web based interface of expert/guideline knowledge modeling tool in order to generate new CDSSs. This tool converts clinical knowledge into CLIPS Rules, and ultimately the knowledge base of the new CDSS is created. To add a CDSS for a disease in PATEL, at first, symptoms of disease (laboratory or non-laboratory) and the range of symptoms are defined (Figure.4). These symptoms are commonly used to define other diseases. In the next step, the clinical rules are entered individually which include rule parameter, rule fact and rule result. For each rule, the rule recommendations for the health care team and the patients are defined and added (Figure.5). These rules are editable.

c. **Operation mechanism of PATEL:** In the following, we explain the operational process of PATEL to CDSS-based Tele monitoring for different diseases. 2 shows the top-level architecture with the major components inside the CDSSs Middleware.

- **User's Request:** Patients through a graphical user interface send their symptoms and parameters. Requests sent as XML files and received by demand management, which is the web service. The web service extracts the necessary parameters for extracting the patient's specific records and for calling the relevant CDSSs. The web service transmits the extracted parameters to the next component via XML files.
- **Patient Data Fetch:** The data access component receives parameters from requests sent by users or sensors. As well as health care team, if they need to have the patient's medical history can send requests through this component. This component is responsible for fetching patient information from EHR.
- **Supervisor:** This component is responsible for the data preprocessing. This component receives extracted parameters from the user request component. Entering such an amount of information to the CDSSs directly increases the speed of the inference process and the speed of the load. The main task of this component is the data classification, data mapping to maintain compatibility with the CDSSs and marking them. It also provides a standard access point for different decision support systems. This component is able to allocate suitable CDSSs to the demand. Depending on the user's facts, Supervisor

will select the appropriate CDSSs for the analysis of the facts. This preprocess is performed by examining the overhead of the data that is sent with the facts. The supervisor also provides interoperability between CDSSs (Figure 3). This means that rule or rules of a CDSS can affect the inference of the rules of another CDSS. Several CDSSs have been used by Multi-Threading programming in the PATEL. It allows to inference facts in CDSSs as parallel.

d. Case Study Evaluation:

The scalability of PATEL shows that it can extend new CDSS through the corresponding custom tools provided by the platform. In order to validate this feature, we uploaded four CDSSs on PATEL. The inference process and the results of four concrete CDSSs related to Complications of diabetes: CDSS for Retinopathy (RETCDSS), Nephropathy (NEPHCDSS), Neuropathy (NEUCDSS), Cardiovascular disease (CARDCDSS) are illustrated as following. According to the working mechanism of PATEL, the first effort of customization is configuring the clinical knowledge by the web base interface of expert/guideline knowledge modeling tool in order to generate new medical knowledge data for CDSSs. These symptoms and their range (High, Medium, and Risk) are once defined and commonly used for all diseases and their related CDSSs (Figure 4).

Knowledge of different diseases were added into different CDSS based on PATEL web based tool. Part of the rules of the NEUCDSS is shown in Figure 5. To verify the inference of CDSSs and interoperability between CDSSs and EHR, a scenario defined for a diabetic patient. Initially, the physician, after creating a user account for the patient, identified in his panel the symptoms and parameters (sensation of touch and sight, blood pressure, etc.) that the patient should measure and send at a specific time at the home.

e. Performance evaluation:

An important factor in any CDSS is performance. A common goal among the developers of CDSSs is the response time based on the sub-second[14]. In PATEL platform, since having more than one CDSSs, the performance of the platform in terms of Response Time is an important factor in performance measurement. The PATEL platform has four types of delays: transmission delay (The time needed to transmit a message over the network once latency is overcome)[6], Patient data fetch from EHR and Parsing overhead (Identify and select CDSSs commensurate with symptoms and parameters). Formula 1 shows Response Time computation.

Response Time =

$$= \sum \textbf{Transmission delay}. \textbf{Patient data fetch}. \textbf{Parsing overhead}. \textbf{Inference time}$$

Formula 1. Response Time calculation

To evaluate the robustness of the PATEL platform and its associated delay, we conducted an evaluation based on time. A test system designed for the purpose that send requests to the servers at different times of the day for three days. In total 800 requests were sent to the servers. These requests automatically generated various values of the symptoms and parameters of patients with diabetes conditions (CDSSs of Retinopathy, Nephropathy, Neuropathy, and Cardiovascular disease uploaded on the PATEL at first). This requests sent to servers and then patient health conditions were inference. Finally, delays and response time of this process calculated.

Results:

a. **Case study result**: According to the data available in the EHR (Albuminuria+, HTN+, a 7-year history of diabetes and Last visit to ophthalmologist 6 months ago) and patient-reported data (Blood pressure:140/90mmgH, heat loss, Blurry vision) (Fig.6), RETCDSS, CARDCDSS and NEUCDSS were evaluated the condition of the patient as Warning, NEPHCDSS was evaluated conditions as Critical.

Comparing the platform results with the specialist physician opinion about the patient's condition, the correct conclusion of the platform, and the relationship between the decision support systems of the NEPHCDSS and the CARDCDSS (due to the effect of blood pressure on the kidney and heart condition) and the interaction of CDSSs with the history of patient information in the EHR was confirmed (Figure 7).

By selecting the statuses declared by each of the CDSSs, the health care team can view additional information such as patient medical records, patient-submitted information, clinical rules that produce results and recommendations to the health care team (Figure 8). Based on Figure 8, additional information on patient status (warning) based on NEUCDSS is shown.

b**. Accuracy evaluation result**:

To evaluate the accuracy of the presented platform 250 patient records were used. 55 cases were Retinopathy, 72 cases were Cardiovascular, 64 cases were Neuropathy and 59 cases were Nephropathy. The data in these patient records were uploaded to the platform using a Mobile-based interface. Patients' conditions were evaluated using clinical decision support systems available on the platform. According to Table 1, of the 59 cases evaluated for Retinopathy, 53 were correctly diagnosed. In 72 Cardiovascular cases, condition of 63 was correctly diagnosed. In 64 neuropathy cases, condition of 59 were correctly diagnosed. In 59 Nephropathy cases, 54 patients reported the correct condition. The platform precision in three conditions (normal, warning and critical) for retinopathy, cardiovascular, neuropathy, and nephropathy are shown in Table 1.

Table 1. Platform accuracy result

Based on this evaluation, the platform presented in this article was successful in diagnosing 91.6% of the cases.

c. **Performance evaluation result**:

The delay in transmission is significantly dependent on the distance between the sender and the server. This time usually less than a millisecond in a subnet. Since the amount of data transfer in the architecture of this article is small, the transmission delay will be very small and will be between 10 and 90 milliseconds (MSc), respectively. The time taken to fetch patient data from EHR is considerably quicker (between 90-150 MSc). PATEL is established on Extensible Markup Language (XML) protocols and there is overhead in parsing the XML. Given the size of the messages used in PATEL, the overhead is honestly minor. The overhead time for messages in PATEL architecture was approximately 11 MSc. The last source of delay is the inference time. Simple requests can make their conclusions very fast (between 50 and 100 MSc), but other request use roughly twice as much time. The mean response time for the CARDCDSS is 491MSc and for the NEPHCDSS is 614 MSc, for the NEUCDSS is 512 MSc and for the RETCDSS is 556 MSc, with a workload equal to 100 requests (Table 2).

Table 2. Estimation of PATEL efficiency based on 100 requests per day (Avg Time/MSc)

In Table3, Patel architecture performance evaluation is calculated and displayed for 200 workloads. The mean response time of the CARDCDSS is 541 MSc. The mean response time for the NEPHCDSS is 687 MSc, for the NEUCDSS is 754 MSc and for the RETCDSS is 523 MSc.

Table 3. Estimation of PATEL efficiency based on 200 requests per day (Avg Time/MSc)

Also in Table4, the evaluation of proposed architecture performance is calculated and displayed based on 500 workloads. The mean response time of the CARDCDSS is 584 MSc. The mean response time for the NEPHCDSS is 780 MSc, for the NEUCDSS is 660 MSc and for the RETCDSS is 468 MSc.

Table 4. Estimation of PATEL efficiency based on 500 requests per day (Avg Time/MSc)

These results indicate that based on 100, 200 and 500 requests, sent to the server over 24 hours for three days, The Patel architecture is very reliable and fast, and it is useful for supporting various types of CDSSs. Based on the evaluation of the work load, which contained 100 requests, the minimum time and maximum time for transmission delay, patient data fetch, parsing overhead, and inference time were examined (Table 5).

Table 5. Minimum and Maximum Time (MSc) for Transmission Delay, Patient Data fetch, parsing overhead, and Inference Time Based on 100 requests

According to Table5, the minimum and maximum time of four criteria delay in the PATEL are calculated. The inference time has the highest delay, and the latency of the parsing overhead is the least delayed. Meanwhile, fetching patient information from EHR also had a significant delay, affecting the response time.

Discussion:

The work describes the software platform. The scalability of PATEL can comfort admin to upload new CDSSs for different diseases easily without having computer programming and coding skill. The platform Provides interoperability between CDSSs, if there is a relationship between them and can interoperate with EHR. Proposed platform has potential for generating and supporting a commercial market for CDSSs. As it developed based on modules, allowing the platform to be distributed. PATEL will reduce cost and time of trying new CDSSs because of its ability to easily integrate a variety of CDSSs, and to easily edit them. PATEL is based on the principles of a SOA. One of the evaluation benchmarks considered in the architectures of CDSSs based on SOA is response time [15]. In the PATEL platform, 89 percent of the requests were answered in less than one second and precision of 91.6 percent. one of the methods used to evaluate scalability is case study[15].To evaluate the scalability of the PATEL, four clinical decision support systems were selected for use and validation of this architecture. The results confirm the correct functioning of the platform. The proposed platform can support CDSSs of various diseases simultaneously and provides the necessary scalability to add a new CDSS. Tele-monitoring systems will be capable of service by connecting to this platform.

Conclusion:

The most important feature of the proposed platform that distinguishes it from other existing architectures is its scalability. It is possible to upload pre-created knowledge bases or using the management panel to create a new knowledge base. Different health systems based on decision support systems will be able to serve as a connection to this collaborative infrastructure and it is expected that using this infrastructure will lead to a lot of rework in the implementation of these systems. As a limitation, in designing this platform, a rule-based technique has been used as a method of knowledge representation, which is not supported techniques such as semantic network, frames, and logic as a limitation. We will remove this limitation in the future.

Declarations

Availability and requirements:

Project name: PATEL: "Designing, implementing, and evaluating a clinical decision support system Platform based on service-oriented architecture".

Project home page: The platform link was on the hospital local server so it is not accessible.

Operating system(s): Android, Linux

Programming language: CLIPS,PHP, Google Android Studio

Other requirements: PHP-CLIPS plugin.

License: license is not needed.

Any restrictions to use by non-academics: nothing.

Ethical approval and consent: This article is taken from the thesis entitled "Designing, implementing, and evaluating a clinical decision support system Platform based on service-oriented architecture" in Doctor of Philosophy(PhD) of Medical Informatics in the School of Allied Medical Sciences of Tehran University of Medical Sciences. Evaluated by: School of Public Health & Allied Medical Sciences- Tehran University of Medical Sciences, Approval Date: 2019-03-06; Approval ID: IR.TUMS.SPH.REC.1397.296.
URL:http://ethics.research.ac.ir/ProposalCertificateEn.php?id=53581&Print=true&NoPrintHeader=true&NoPrintFooter=true&NoPrintPageBorder=true&LetterPrint=tru

Consent for publication: Not applicable.

Availability of data and materials: The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Competing interests: The authors declare that they have no competing interests.

Funding Information: Tehran University of Medical Sciences (grant number: 9421481001). The funder paid for the study. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Authors 'contributions: all authors have read and approved the manuscript. AY, RS, RSH contributed equally to the study design. AZ and MZ prepared the manuscript and MZ revised it critically. AZ was the initiator of the literature search and review, reading, categorizing and analyzing and developing the proposed platform. RS and RSH additionally performed supervisory tasks.

Acknowledgements: Not applicable.

References

1. Power, D., *Decision support systems: From the past to the future.* AMCIS 2004 Proceedings, 2004: p. 242.
2. Middleton, B., D. Sittig, and A. Wright, *Clinical decision support: a 25 year retrospective and a 25 year vision.* Yearbook of medical informatics, 2016. **25**(S 01): p. S103-S116.
3. Blanchard, B.S., W.J. Fabrycky, and W.J. Fabrycky, *Systems engineering and analysis*. Vol. 4. 1990: Prentice Hall Englewood Cliffs, NJ.
4. Yazdani, A., et al., *Scalable architecture for telemonitoring chronic diseases in order to support the CDSSs in a common platform.* Acta Informatica Medica, 2018. **26**(3): p. 195.
5. Zhang, Y.-F., et al., *Design and development of a sharable clinical decision support system based on a semantic web service framework.* Journal of medical systems, 2016. **40**(5): p. 118.

6. Loya, S.R., et al., *Service oriented architecture for clinical decision support: A systematic review and future directions.* Journal of medical systems, 2014. **38**(12): p. 140.
7. Rodriguez-Loya, S. and K. Kawamoto, *Newer architectures for clinical decision support*, in *Clinical Decision Support Systems*. 2016, Springer. p. 87-97.
8. Weber-Jahnke, J., *Design of decoupled clinical decision support for service-oriented architectures.* International Journal of Software Engineering and Knowledge Engineering, 2009. **19**(02): p. 159-183.
9. Kawamoto, K., et al. *Clinical information system services and capabilities desired for scalable, standards-based, service-oriented decision support: consensus assessment of the Health Level 7 clinical decision support Work Group*. in *AMIA Annual Symposium Proceedings*. 2012. American Medical Informatics Association.
10. Dobrescu, R. and V. Purcarea, *Impact of information technology on the quality of health services*, in *Service orientation in holonic and multi-agent manufacturing control*. 2012, Springer. p. 307-319.
11. Nannings, B. and A. Abu-Hanna, *Decision support telemedicine systems: A conceptual model and reusable templates.* Telemedicine Journal & e-Health, 2006. **12**(6): p. 644-654.
12. Chu, H., et al. *A scalable clinical intelligent decision support system*. in *International Conference on Smart Homes and Health Telematics*. 2016. Springer.
13. Oertel, G.F., *The barrier island system*. 1985, Elsevier.
14. Doherty, W.J. and A.J. Thadhani, *The economic value of rapid response time.* IBM Report, 1982.
15. Wright, A. and D.F. Sittig, *SANDS: a service-oriented architecture for clinical decision support in a National Health Information Network*. Journal of biomedical informatics, 2008. **41**(6): p. 962-981.

Tables

Table 1. Platform accuracy result

Critical precision	Warning precision	Normal precision	Correct diagnosis of the patient's condition	Number of records	Disease
91,7%	93,4%	85%	53	59	Retinopathy
88,2%	86,2%	88,5%	63	72	Cardiovascular
95,2%	85,7%	90,1%	59	64	Neuropathy
90%	94,7%	89,4%	54	59	Nephropathy

Table 2. Estimation of PATEL efficiency based on 100 requests per day (Avg Time/MSc)

CDSS	Request Number	Transmission delay	fetch patient information	Parsing overhead	Inference Time	Response Time	Standard deviation
CARDCDSS	22	81	131	6	273	491	89
NEPHCDSS	29	88	148	11	369	614	116
NEUCDSS	23	75	137	6	294	512	97
RETCDSS	25	84	141	9	322	556	105

Table 3. Estimation of PATEL efficiency based on 200 requests per day (Avg Time/MSc)

CDSS	Request Number	Transmission delay	fetch patient information	Parsing overhead	Inference Time	Response Time	Standard deviation
CARDCDSS	48	89	164	7	281	541	119
NEPHCDSS	51	93	189	10	395	687	134
NEUCDSS	55	109	201	11	433	754	145
RETCDSS	46	84	158	6	275	523	104

Table 4. Estimation of PATEL efficiency based on 500 requests per day (Avg Time/MSc)

CDSS	Request Number	Transmission delay	fetch patient information	Parsing overhead	Inference Time	Response Time	Standard deviation
CARDCDSS	115	94	185	9	296	584	128
NEPHCDSS	191	105	213	16	446	780	151
NEUCDSS	128	101	199	11	349	660	139
RETCDSS	66	55	107	7	233	468	93

Table 5. Minimum and Maximum Time (MSc) for Transmission Delay, Patient Data fetch, parsing overhead, and Inference Time Based on 100 requests

Criteria delay time	Minimum Time	Maximum Time
Transmission Delay	9	100
Patient Data fetch	90	160
Parsing overhead	4	16
Inference Time	46	480

Figures

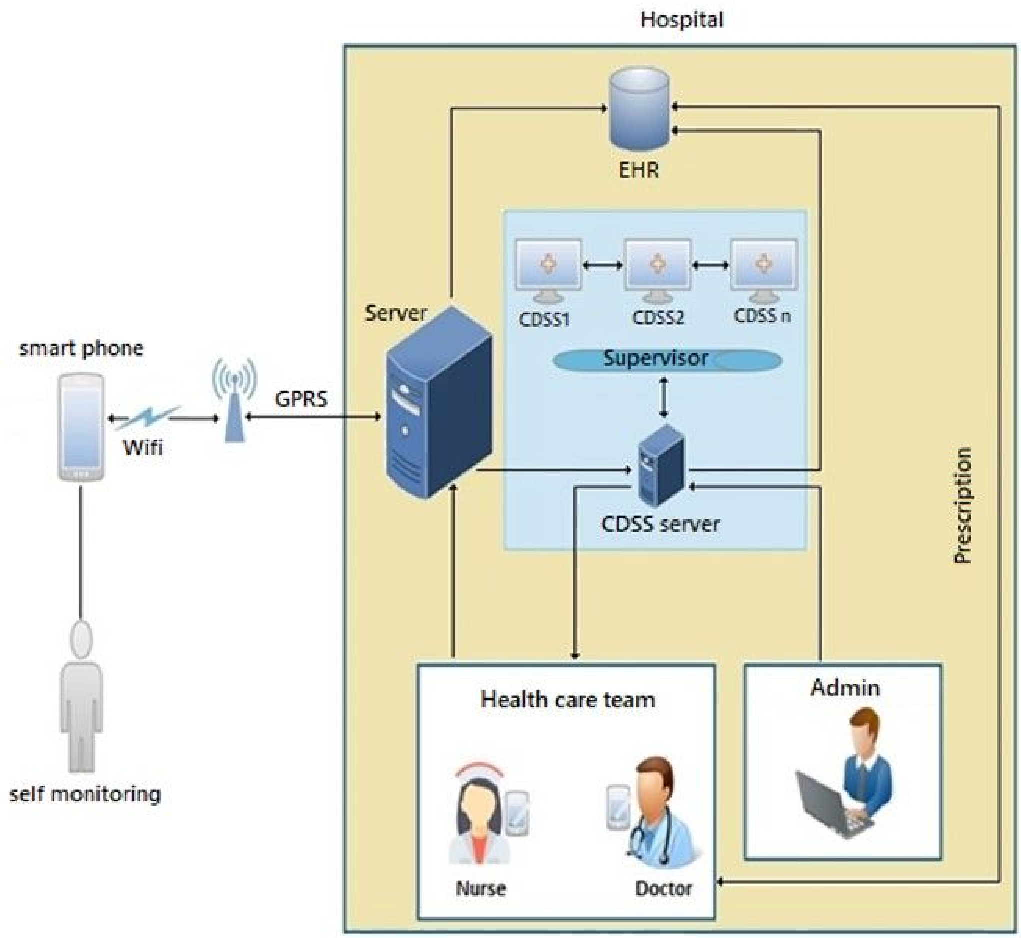

Figure 1

Shows the high-level architecture with the main components of the platform PATEL.

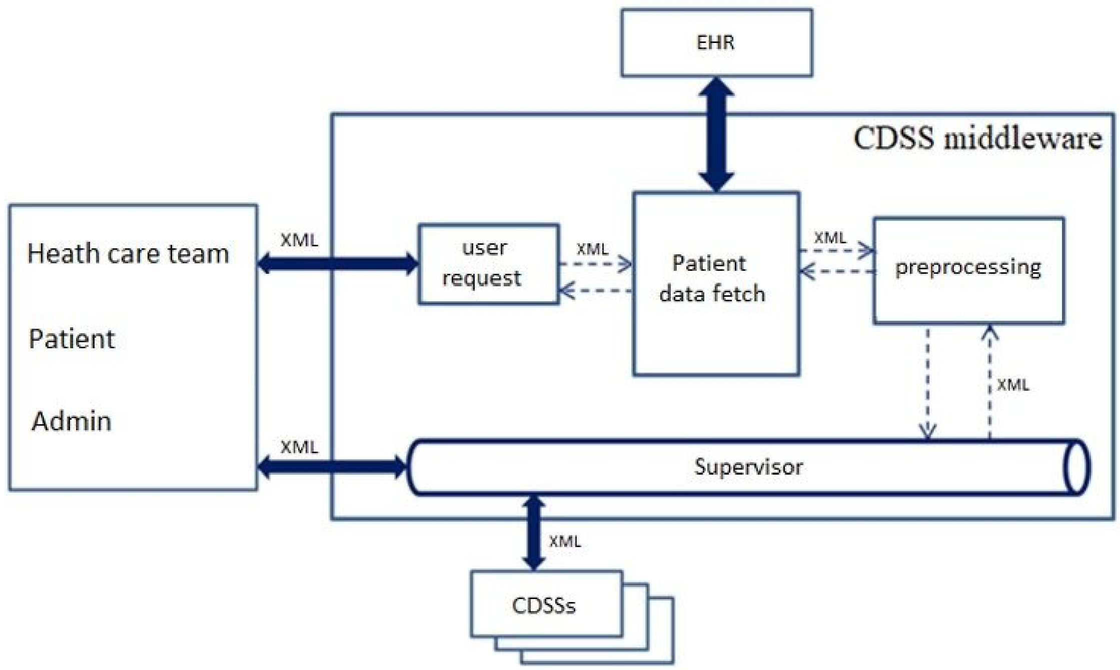

Figure 2

the Integration platform with CDSSs Middleware

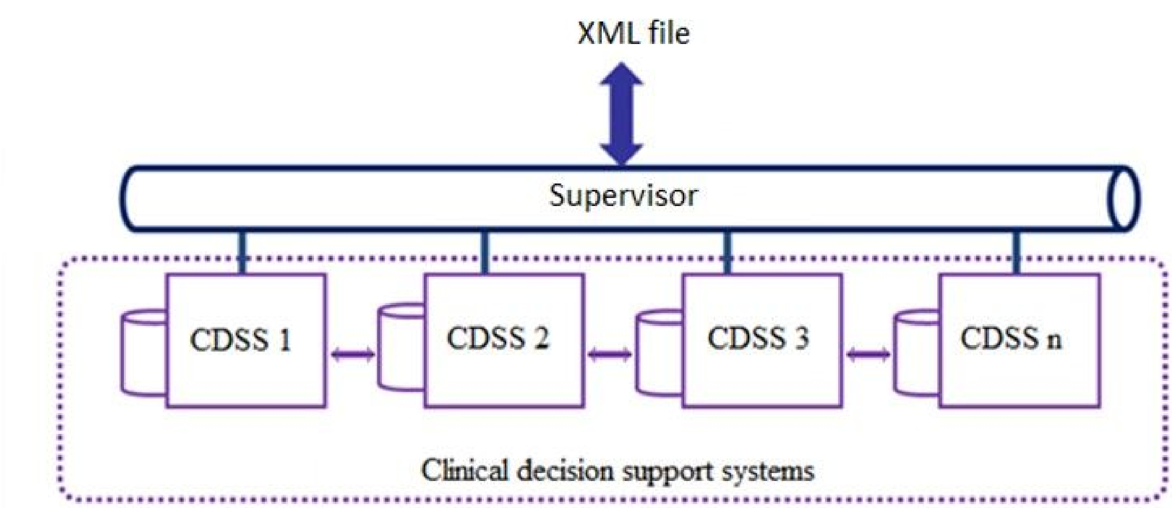

Figure 3

the Schematic Architecture of CDSSs AND Supervisor

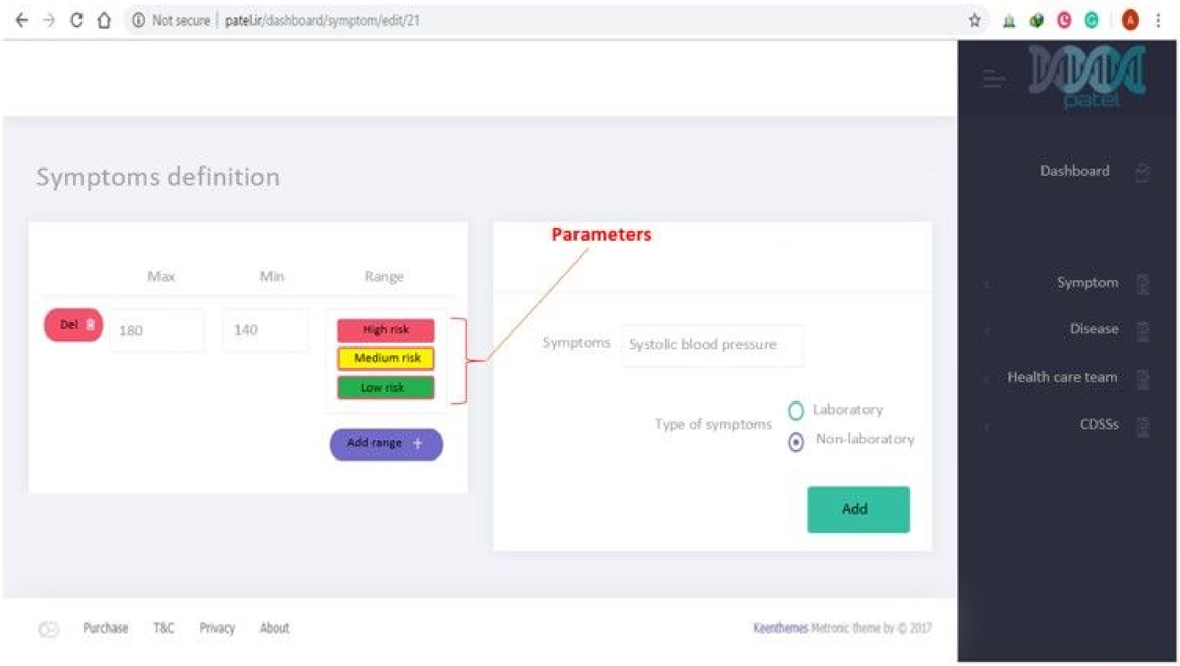

Figure 4

Definition of disease symptom and their ranges

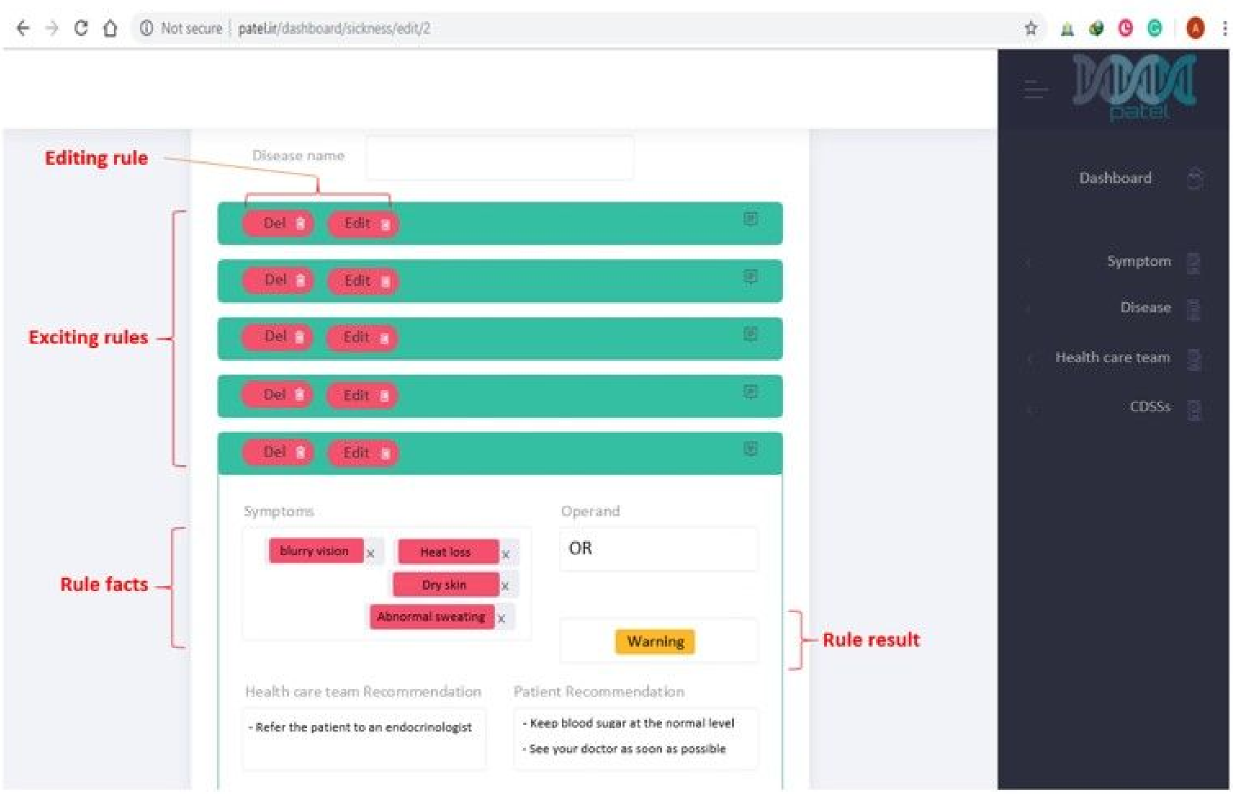

Figure 5

Definition of rule fact, rule result and recommendation for patient and health care team.

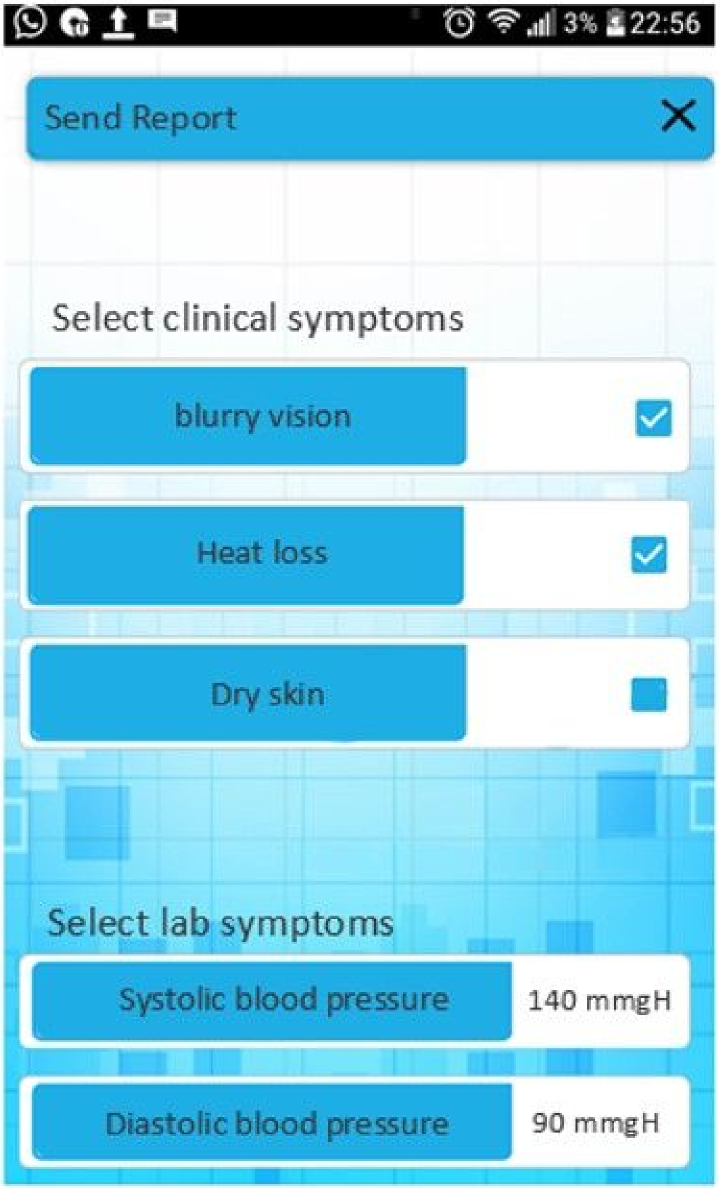

Figure 6

Patient interface and data entry

Figure 7

CASE –Study results: Diagnosis of the patient's condition (health care team interface)

Figure 8

Additional patient report (health care team interface) based on NEUCDSS