# Development of Data Storage and User Interface in the Clinical Decision Support System

G. R. Shakhmametova[1](✉), A. A. Evgrafov[2], and R. Kh. Zulkarneev[3]

[1] Ufa University of Science and Technology, K. Marx Str. 12, Ufa, Russia
shahmametova.gr@ugatu.su
[2] "GPN" - Digital Solutions, Revolyutsionnaya Str. 96/4, Ufa, Russia
[3] Bashkir State Medical University, Lenin Str. 3, Ufa, Russia

**Abstract.** This article discusses ways of storing information processed in the developed clinical decision support system (CDSS) in the diagnosis of respiratory diseases. An overview of the information and analytical system "ProMed", used in medical institutions today, from the point of view of storing information, is provided, along with comparative analysis of existing solutions in the field of DBMS, such as Oracle, MySQL, PostgreSQL, SQLite, MongoDB, Microsoft SQL. Their advantages and disadvantages in relation to the developed DSSP, as well as the choice of PostgreSQL as a DBMS is justified. The structure of the stored data is also shown as a list of tables; dependencies and links are shown. Additionally, a layout of the interface of the solution being developed is shown, including forms for viewing the patient database and searching for individual records, adding/changing/deleting a patient record, working with statistical indicators, diagnostics and clinical recommendations. #CSOC1120.

**Keywords:** Clinical Decision Support System · Data Processing · Data Storage · Database · Application Interface

## 1 Introduction

Information technology increases its importance for medicine and health care every day. Methods of intellectual analysis and information processing are becoming widespread, which ultimately leads to the creation of clinical decision support systems (CDSS) [1]. Systems of this type are intended to assist doctors and other medical professionals in working with tasks related to making clinical decisions [2].

Previously, the authors in [3, 4] described the development of the architecture of the clinical decision support system in the diagnosis of respiratory diseases (see Fig. 1). This system assumes the presence of such functions as the analysis of patient data in the form of textual, graphic and sound information, the formulation of a presumptive diagnosis based on the analysis and the identified patterns, and the formation and issuance of medical recommendations to improve the effectiveness of patient treatment.

The information processed by the developed CDSS includes:

1. Text data - presented in the form of medical records expressed in natural language, test results, patient case histories, etc. in the format of scanned images.


R. Silhavy and P. Silhavy (Eds.): CSOC 2023, LNNS 722, pp. 808–816, 2023.
https://doi.org/10.1007/978-3-031-35311-6_75

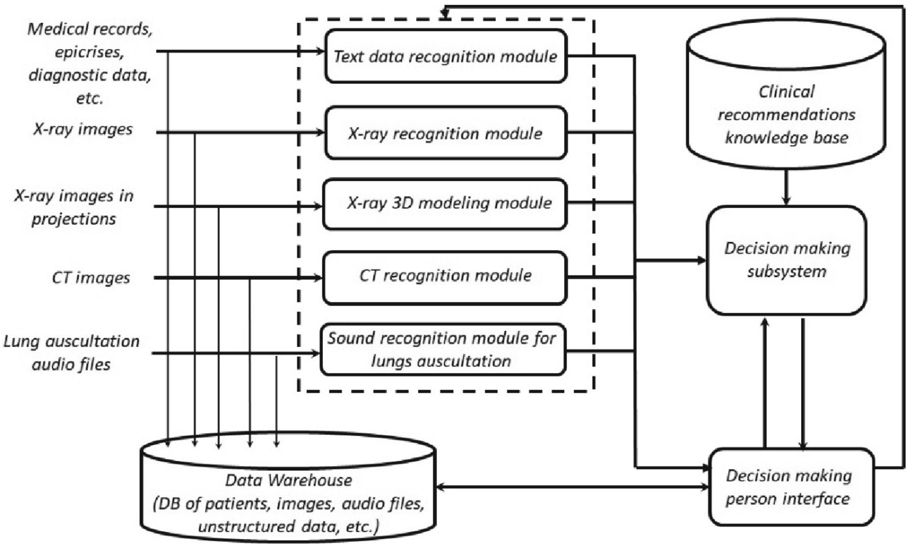


**Fig. 1.** Architecture of the developed CDSS

2. Graphic data - presented by blocks of recognition of X-ray images and images of magnetic resonance imaging.
3. Sound data - provided by the sound recognition unit during auscultation of the lungs.

The principles of processing textual and graphical data in CDSS in the diagnosis of respiratory diseases were considered by the authors in [5] and [6]. This article is devoted to the organization of data storage and development of the interface of the CDSS prototype.

Analysis and processing of medical textual and graphic information, in turn, are quite complex and resource-intensive due to the large amount of processed data [7]. For a comfortable interaction with the CDSS, such processes should not have a negative impact on the quality and speed of services, and the issues of storing and presenting data require separate consideration, which is what this article is devoted to. Section 2 examines the information and analytical system “ProMed” that is used in medical institutions, lists its advantages and disadvantages, and highlights the development vector that requires special attention in the development of the CDSS. In the third section, a solution for storing the processed information is proposed, and the implementation of the interface for the software product is given.

## 2 Problem Definition

In the modern world, processing of large amounts of information inevitably requires its structured storage and convenient presentation to the end user.

Every day, medical personnel analyzes and processes colossal amounts of patient data in the course of diagnostics, disease management and treatment [8]. For a long time, information was stored in paper-based medical records; in recent decades, the electronic form of data storage has become widespread. In particular, in medical institutions the regional information and analytical medical system “ProMed” is utilized [9]. The main advantage of this system is the ability to maintain an electronic medical record of the patient (EMR) - a complex of medical data on the patient’s health status and the treatment prescribed to him. The EMR stores information about each case of a patient’s visit to a polyclinic, a hospital, or an ambulance service. Thanks to the formation of a unified regional electronic medical record, a doctor from any medical organization in the region quickly gets access to data on the patient’s health, regardless of where the patient was treated earlier, with exception for diagnoses in such areas as drug abuse, psychiatry, STD, AIDS, or tuberculosis. At the same time, “ProMed” has one significant drawback - the data is stored in it mainly in an unstructured form. This is due to the lack of uniformly established standards for medical reporting when working with patients, as well as a variety of technical equipment and many types of procedures in various clinics. The indicated drawback is not significant when it comes to interaction with one patient, or with different patients within the same medical institution. However, a wide diversity of forms of preserved medical documents in “ProMed” leads to significant difficulties in cases when there is a need for statistical processing of previously obtained information in order to conduct any clinical research. This aspect requires additional analysis and should be taken into account when developing a CDSS in the diagnosis of respiratory diseases.

The second point requiring close attention when developing a CDSS is the interaction with the user interface. The interface should be responsive and intuitive to use in order to reduce the burden on medical staff and improve the efficiency of medical care.

## 3 Proposed Solution

The first step is to determine how the data is stored. The simplest is storage in formats such as bin, txt, doc, as well as in spreadsheets such as Excel. However, this method is not suitable for storing and processing a substantial amount of information. In such cases, databases prove to be the most efficient solution.

During the development process, an analysis of existing databases for their advantages and disadvantages was carried out. Conclusions of this analysis are presented in Table 1. For a comparative analysis, the most famous DBMS were selected, such as Oracle, MySQL, PostgreSQL, SQLite, MongoDB, Microsoft SQL.

Oracle Database [10] is one of the most widely known object-relational database management systems. It implements technologies inherent in object-oriented DBMS and utilizes an object-oriented approach based on classes and inheritance. Oracle Database comes in six different editions to support a variety of application development and deployment scenarios (with a large variability in pricing for its products). The use of this DBMS is complicated by paid distribution, while the requirements for the functional component of the DBMS can be implemented on freely distributed analogues. Another disadvantage of Oracle relative to its competitors is the high requirements for computing resources allocated to the DBMS.

MySQL [11] is an open-source relational database management system with a client-server model. The flexibility of MySQL is supported by a large number of table types:

**Table 1.** Comparative characteristics of the studied databases

| Database name | Dignity | Disadvantages |
|---|---|---|
| Oracle | Big data processing capability | Paid distribution |
| MySQL | Free distribution, ease of use | Limited functionality, low reliability |
| PostgreSQL | Free distribution, scalability | High complexity of customization |
| SQLite | Consists of one file, included in the Python package | Lack of opportunity to increase productivity |
| MongoDB | Free version, high speed, ease of use | SQL is not used as a query language |
| Microsoft SQL | Simplified interoperability with other Microsoft products | High cost, significant resource consumption |

users can choose from both MyISAM tables which support full-text search, and InnoDB tables, which support transactions at the level of individual records. Moreover, MySQL contains a special EXAMPLE table type that demonstrates how to create new table types. Thanks to its open architecture and GPL licensing, new types of tables are constantly appearing in MySQL. An important advantage is also the presence of MySQL in the standard repositories of many Linux distributions, which greatly simplifies the initial installation. However, MySQL was not chosen as a DBMS for CDSS due to lack of reliability, which is one of the critical factors when working with personal data. In case of incorrect shutdown of the database, the risk of errors in the data is high. Also, MySQL has a rather low performance compared to its counterparts, which can be critical when used in CDSS, and only paid technical support is available in the free version.

PostgreSQL [12], in turn, is based on the nonprofit Postgres database developed as an open-source project at the University of California, Berkeley. Available in implementations for a variety of UNIX-like platforms, including AIX, various BSD systems, HP-UX, IRIX, Linux, macOS, Solaris/Open Solaris, Tru64, QNX, and Microsoft Windows. It is one of the most stable DBMS, and the high complexity of configuration is compensated by the speed and flexibility of work. PostgreSQL is a scalable solution that allows you to process terabytes of data if necessary. There are also many predefined functions that facilitate calculations (including the ability to write your own functions). This DBMS meets all the requirements for the development of a CDSS.

SQLite [13] is an embedded cross-platform database that supports a fairly complete set of SQL commands and is available in source code (in C). Since it is an exchange protocol, calls to functions (API) of the SQLite library are used. This approach reduces overhead and response time, and simplifies the program. SQLite stores the entire database (including definitions, tables, indexes, and data) in a single standard file on the computer where the program is running. It is easy to implement, because, before the start of the execution of the write transaction, the entire file storing the database is locked. SQLite is rather limited in functionality and extensibility, and therefore its use in the system under development is considered inappropriate, and as such, the use of the database is possible only for educational purposes or for prototyping.

MongoDB [14] is a document-oriented database management system that does not require a description of the table schema. Considered one of the classic examples of NoSQL systems, it uses JSON-like documents and a database schema. Written in C++, it is used in web development, in particular, within the framework of the MEAN JavaScript-oriented stack. The use of MongoDB is difficult due to the lack of support for the SQL language used as queries. Databases that support a relational data model will be more efficient in the context of the task of creating a CDSS, since the task implies a lot of data with a lot of relationships.

Microsoft SQL Server [15] is a relational database management system developed by Microsoft Corporation. The main query language used is Transact-SQL, a joint development of Microsoft and Sybase. Transact-SQL is an implementation of the ANSI/ISO Structured Query Language (SQL) standard with extensions. It is designed to work with databases ranging in size from personal to large enterprise-scale databases. One of the advantages is the presence of an efficient garbage collector, whose work fits into 15–20% overhead, which is less than its counterparts. The use of Microsoft SQL Server in CDSS is not advisable due to the high cost and special adaptation for Microsoft products (these requirements are not imposed on the CDSS). As a result of the analysis, PostgreSQL was chosen as a DBMS for the developed CDSS because of its high power and scalability in comparison to other free DBMS.

Since the storage of information in the database is possible only in tabular form, the CDSS, in the process of work, carries out the following preliminary processing of information before placing it in the database:

1. Text data are analyzed and useful information is extracted from them in accordance with a known data format stored in the database.
2. Graphic and sound data are not stored directly in the database; tables containing paths to files are used to access information of this type.

The list of generated tables is shown in Fig. 2. Tables with names beginning with "main" are the main tables used in the course of the CDSS. Tables labeled "auth" are intended for distribution of access rights and user authorization. Also, additional internal tables with the "django" label are formed, containing information about the structure of the created tables, migrations and logging.

Figure 3 shows data streams processed in the developed CDSS. The main table is the Patients table, which accumulates patient data. This table is adjoined by 5 tables, including the analysis data of each of the patients, if any ("Biochemical blood test", "Clinical urine test", "Coagulogram test", "Blood test", "Sputum test"). If necessary, the list of tables of this type can be expanded. In the course of the work of the CDSS, the data generated for the analysis fall into the temporary cluster "Analysis processing", which also uses the knowledge base and rules ("Medical information source, rules") for the correct processing of data and diagnosis of patients. The results are stored in the "Work results" table and can be output from the system in the form of text documents.

The choice of the graphical interface was carried out from the 3 most optimal options:

1. Window application, interface implemented in C++/C# languages.


**Fig. 2.** Tables generated in the CDSS database

2. Window application, interface implemented in Python languages.
3. A web interface implemented in Python.

Due to the fact that the main part of the system is implemented in the Python language, and is also not demanding on the wide possibilities of window graphics, the choice was made in favor of the web interface in Python. Django is used as a framework for a web application [16].

The reason for choosing Django is due to a number of advantages:

– availability of own ORM for interaction with the database.
– built-in interface for administration, authorization and authentication.

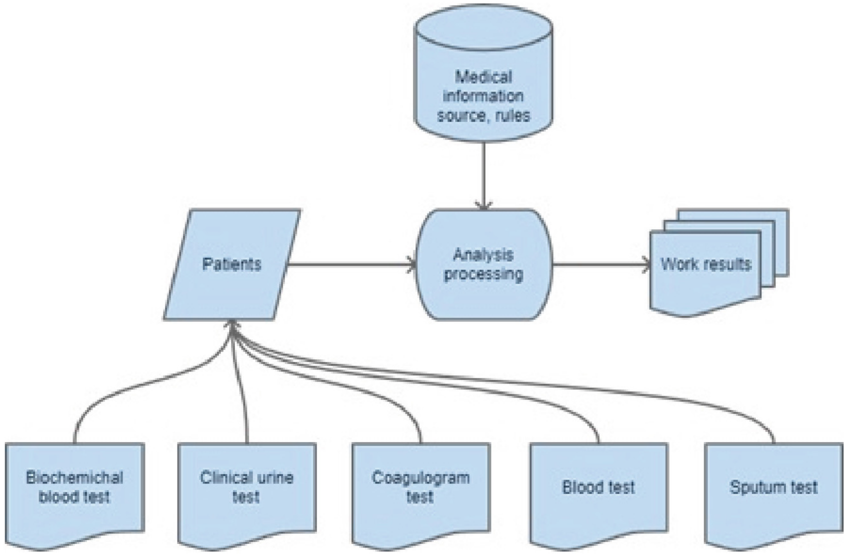


**Fig. 3.** Data streams implemented in the CDSS

– a convenient library for working with forms, providing the implementation of basic templates for generating the application interface.

With a significant increase in the number of users of the SPKR, a transition from Django to FastAPI is allowed due to greater asynchrony, which will allow processing a larger number of clients at a time.

The pre-designed web application interface contains the following forms:

1. The form of viewing the patient database and searching for individual records - the main form of the program, navigates through the patient database, allowing you to search for individual patients, if necessary, by patient ID, partial or full name, additional information about the patient.
2. The form for adding/changing/deleting a patient record - interacts with a single record containing information about a single patient. The add form is activated by clicking "Add" on the main form screen, and the form for editing information about an existing patient (including deleting it) by selecting a specific patient from the list of patients and then clicking "Edit" or "Delete".
3. The form of work with statistical indicators includes a complex of formed data on the average and median indicators of samples among patients according to key indicators, such as the number of days since the onset of the disease, the presence of certain symptoms, the range of results of the analyzes performed. This form includes the ability to build summary tables and graphs of parameters that are of interest to the user of the CDSS, as well as their filtering and the ability to present as an overlay of several parameters on a single graph, taking into account the difference in scaling for the purpose of visual comparison.

4. The form of diagnosis and clinical recommendations implements the main functionality of the DSS, it displays the results of the work carried out with an individual patient; in particular, the options for possible diagnoses established by the system are displayed, with a probabilistic assessment, as well as clinical recommendations generated based on each diagnosis with the participation of the knowledge base.

The layout of the main application window is shown in Fig. 4.

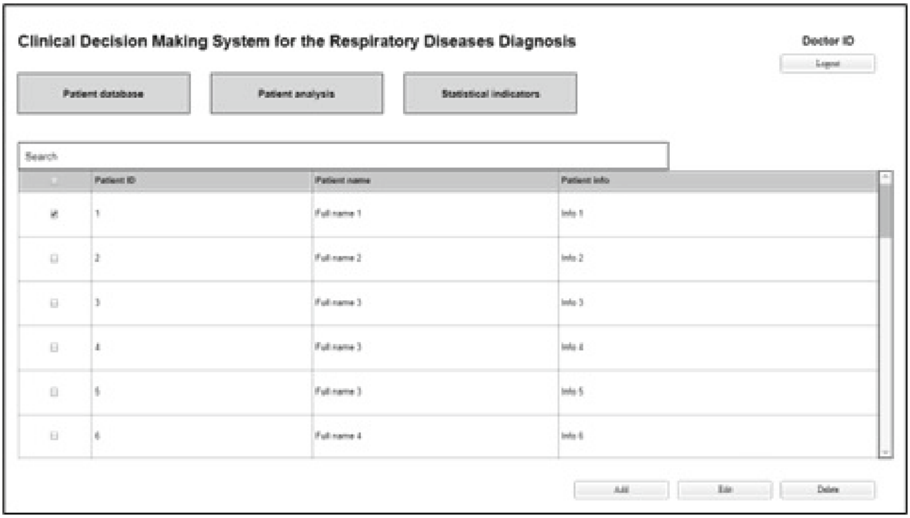


**Fig. 4.** User interface

The layout shows the form for viewing the patient database and searching for records. Each cell contains a record about an individual patient, including his ID, full name and brief information. The "Search" window allows you to search by the "Patient ID" or "Patient name" field. Also, from this form, you can go to the tabs "Patient analysis" and "Statistical indicators". The layout was implemented in the Pencil application, frontend development in CSS and HTML.

## 4 Conclusion

The choice of a database for storing information in the CDSS is one of the fundamental issues in its development. A DBMS that fully realizes its advantages is capable of providing secure storage, redundancy and prompt access to the required information.

The interface, in turn, is also an important link in the decision support system in the diagnosis of respiratory diseases, since it provides a comfortable interaction between the user and the system.

The article discusses the optimal methods of storing information, presents analysis and selection of a database that meets the requirements of the developed CDSS, and also makes a choice and shows the implementation of the system's web interface. The main advantage of the developed solutions for storing the processed information in the CDSS is the possibility of their application and expansion for any other types of diseases.

Further research involves checking performance and speed of the CDSS analysis in the diagnosis of respiratory diseases on a large amount of stored data, as well as further development of the interface, taking into account the recommendations of the medical staff.

**Acknowledgment.** The reported study was funded by RSF, project number 22-19-00471.

## References

1. Sutton, R.T., Pincock, D., Baumgart, D.C., et al.: An overview of clinical decision support systems: benefits, risks, and strategies for success. NPJ. Digit. Med. **3**, 17 (2020). https://doi.org/10.1038/s41746-020-0221-y
2. Buchard, A., Richens, J.G.: Artificial intelligence for medical decisions. In: Lidströmer, N., Ashrafian, H. (eds.) Artificial Intelligence in Medicine, pp. 159–179. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-64573-1_28
3. Shakhmametova, G.R., Zulkarneev, K.Kh., Evgrafov, A.A.: Clinical decision support system for the respiratory diseases diagnosis. In: Advances in Intelligent Systems Research, ITIDS'2019, vol. 166, pp. 101–106. Atlantis-Press (2019). https://doi.org/10.2991/itids-19.2019.19
4. Bogdanova, D.R., Yusupova, N.I., Zulkarneev, R. Kh.: The concept of a decision support system in the management of treatment and accompaniment of the patient. In: Radek, S. (ed.) Lecture Notes in Networks and Systems, CSOC 2023, Springer, Heidelberg (2023)
5. Shakhmametova, G.R., Zulkarneev, R.Kh., Evgrafov, A.A.: Recognition of text information in the bronchopulmonary diseases diagnosis system. Int. Sci. J. "Industry 4.0" **3**(5)5, 245–247 (2018)
6. Shakhmametova, G.R., Yusupova, N.I., Evgrafov, A.F., Zulkarneev, Kh.: Graphical data processing in the clinical decision making system for the respiratory diseases diagnosis using ML methods. IOP Conf. Ser. Mater. Sci. Eng. **1069**(1), 012009 (2021). https://doi.org/10.1088/1757-899X/1069/1/012009
7. Shortliffe, E.H., Chiang, M.F.: Biomedical data: their acquisition, storage, and use. In: Shortliffe, E.H., Cimino, J.J. (eds.) Biomedical Informatics, pp. 45–75. Springer, Cham (2021).https://doi.org/10.1007/978-3-030-58721-5
8. Cronin, R.M., Jimison, H., Johnson, K.B.: Personal health informatics. In: Shortliffe, E.H., Cimino, J.J. (eds.) Biomedical Informatics, pp. 363–389. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-58721-5_11
9. RIAMS "ProMed" Homepage. https://rtmis.ru/en/glavnaya/. Accessed 03 Mar 2023
10. Oracle Database software product Homepage. https://www.oracle.com/. Accessed 03 Mar 2023
11. MySQL software product Homepage. https://www.mysql.com/. Accessed 03 Mar 2023
12. PostgreSQL software product Homepage. https://www.postgresql.org/. Accessed 03 Mar 2023
13. SQLite software product Homepage. https://www.sqlite.org/index.html. Accessed 03 Mar 2023
14. MongoDB software product Homepage. https://www.mongodb.com/. Accessed 03 Mar 2023
15. Microsoft SQL software product Homepage. https://www.microsoft.com/ru-ru/sql-server. Accessed 03 Mar 2023
16. Django framework Homepage. https://www.djangoproject.com/. Accessed 03 Mar 2023