

UDC 004.223.2

## MAIN ASPECTS DESIGNING A WEB APPLICATION FOR WORKING WITH PATIENT'S MEDICAL DATA

Y. MIANITSKI, D. PASTUKHOV  
Polotsk State University, Belarus

*The article presents the main aspects of architectural features for the design of protected medical information systems. The algorithm of technology of distribution of access rights to medical information is considered. The principles and technologies for designing data storage interfaces are considered.*

**Keywords:** information technology, medical information system, data encryption, data security.

**Introduction.** At the moment, many health care institutions use paper document circulation to store outpatient patient database in the Republic of Belarus. This approach, since the development of information technologies, has been significantly inferior in effectiveness to electronic medical systems for storing and processing patient medical data. In this case, electronic systems have a number of advantages [1]:

- Quality improvement of management of medical, administrative and financial activities in a healthcare organization through the comprehensive introduction of an electronic medical file and an electronic voucher, provision of additional electronic services for automatic self-service for patients;
- Reduction of time required for medical paperwork by obtaining ready-made data from databases and computer input;
- Reduction of time required to report and provide a roster of services based on automatically generated templates;
- Medical staff optimization of the health care institutions;
- Provision of an operational control over the performance evaluations of the clinic, examination of the medical care quality by logging data and preparation of the reporting sample;
- Enhancement of the information reliability about the indicators of treatment and diagnostic work and financial activity using methods of cryptographic protection of information;
- The throughput increase of healthcare institutions by reducing the time spent working with the outpatient file and other documents, and as a result – the reduction of queues;
- Provision of telemedicine opportunities for communication between the doctor and the patient, altogether with a consultation between doctors.

These criteria undoubtedly prove the effectiveness of these systems, as well as the relevance of their development and implementation of such information systems in the health care system.

Based on the above characteristics and requirements of the system, the functional structure of the developed software was designed. It consists of the following subsystems:

1. User authorization subsystem. It is responsible for granting the access rights to data for the current user in accordance with the role, authorization in the app.
2. User profile configuration subsystem. It is responsible for changing the doctors' personal data.
3. Subsystem for working with information. It is responsible for information searching, extracting the necessary information in accordance with a specific request, and conveniently displaying information based on filters for various criteria using the graphical interface.
4. Subsystem for working with a pharmaceutical storehouse. It provides the ability to select a medicine on the basis of symptoms and contraindications. Management and tracking of the information about the status of medicines in the warehouse.
5. Data editing subsystem. It provides the administrator with the ability to edit information about doctors and medical institutions in the app.

**Means of problem solving.** The basis of the developing system is the ability to store, accumulate and process data. Also, in this case, medical data must be stored in a strictly systematic form based on the rules of maintaining medical records and have a clear interrelation.

Based on this data a relational database is to be used to store entities and the SQL query language to build the database. In this case, using a relational database will allow to formally and unambiguously determine the limitations of the database integrity in terms of its conceptual scheme.

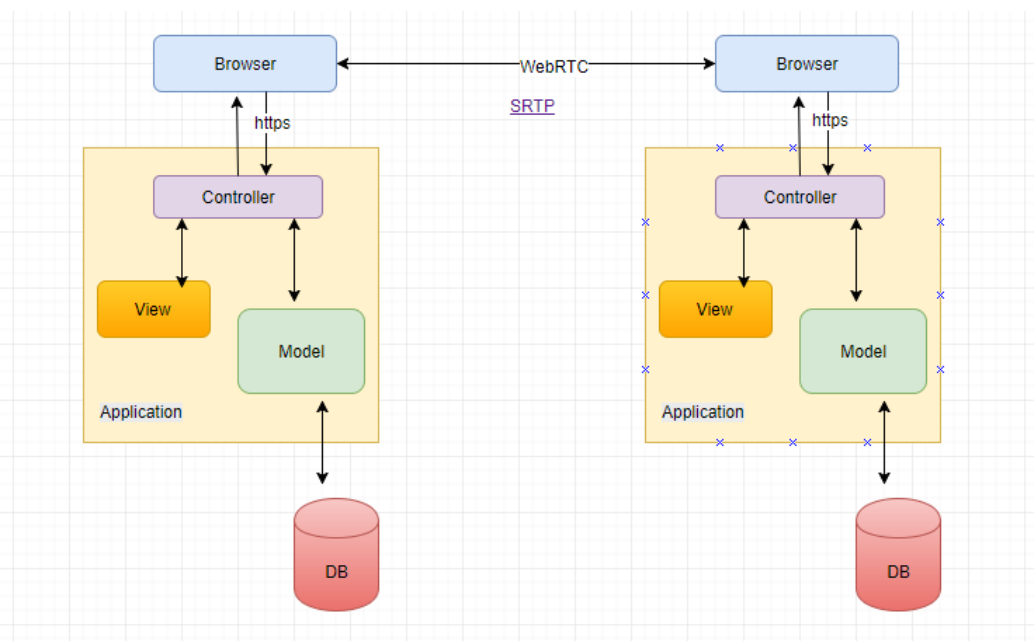
Since medical data is strictly regulated it is advisable to use official documents for drawing up outpatient medical records during system development [2], and present a separate list of the type of examination or data collection as a separate entity associated with the patient.

Obviously it is important to form some architecture when implementing an app. In this case, the MVC pattern is suitable for interaction with the database. It represents an interaction of 3 elements: model, view, and controller [3]. A model is some data that describes a subject area and some operations can be performed on it. This information can be also provided to the user on the graphical interface.

The view is this very graphical interface. Through this interface users of the system will send data to the controller to perform the task. The controller in turn implements some algorithms to get data from the view, process it and send it back.

Also, one of the advantages of such systems is the ability to transmit information in the form of a streaming video message. The implementation of this functionality will allow to conduct video consultations and examinations, which will undoubtedly reduce the time for patient care allowing to unload health care institutions. To implement this system, we will use WebRTC technology.

This is a technology that allows Internet applications to capture video and audio media streams optionally without resorting to intermediary services and protocols. The set of standards that WebRTC technology includes allows data exchange and teleconferencing in node-to-node mode[4].



**Figure 1. - General architectural scheme of the web application for working with the patient's medical outpatient data**

Based on the fact that this system is designed to store confidential patient data it is necessary to develop a system of authentication and cryptographic protection of data.

To provide authentication a user entity in the database with different types of roles, such as patient, doctor, administrator etc. should be created while dividing the access levels to the algorithms for working with data according to the actual level of access to medical documentation. Also, the separation of rights to use algorithms should be implemented both at the database level, through the role editor and through the controller, using verification at the coding stage.

The following technologies are used for secure data transmission over the Internet:

- The HTTPS Protocol is a standard for data transfer between different machines, which defines what should act as a signal for the start of transmission, how data is designated and so on. In this case, data is encrypted using the SSL Protocol, which makes it problematic not only to intercept but also to obtain confidential information[5].
- The SRTP Protocol is a real-time Transport Protocol (RTP) extension profile that appends additional security features such as message authentication, privacy, and listening protection. This Protocol will be used when transmitting video messages [6].
- The AES cipher is a symmetric block encryption algorithm. This type of encryption will use the same symmetric key. This is the best option for the system being developed, since such systems are also protected by legal agreements on non-disclosure to individuals. The encryption method used will be used to encrypt sensitive data at the database level.

**Interface design.** In this application the process of creating an application interface is used for direct interaction between the application and the database combining all the necessary information to output directly to the application. The interface is created by adding the necessary components and then configuring them. The interface of any application has components that allow the program to interact with the user.

This web application is planned to be developed using the Java programming language and the Spring Framework web development framework. It follows that web interface development technologies must be compatible with these technologies, while providing all the features of modern interfaces.

The application being developed by its structure and basic needs should provide opportunities for working and presenting a large amount of data that could be dynamically generated when the user acts. When analyzing tools for creating a web interface, several technologies were identified that are well suited to these goals:

JSP (JavaServer Pages) is a technology that allows web developers to create content that has both static and dynamic components. A JSP page contains two types of text: static source data that can be formatted in one of the HTML, SVG, WML, or XML text formats, and JSP elements that construct dynamic content. In addition, JSP tag libraries as well as EL (Expression Language) can be used to embed Java code in the static content of JSP pages[7].

JSF (JavaServer Faces) is a framework for web applications written in Java. It serves to facilitate the development of user interfaces for Java EE applications. Unlike other MVC frameworks that are managed by queries, the JSF approach is based on the use of components. The state of user interface components is preserved when the user requests a new page, and then restored if the request is repeated. [8]

Since both technologies support the ability to work with dynamic data, the choice of JavaServer Pages technology was given because its support is most intensive and this technology is more compatible with the Spring Framework due to working through Servlet.

It was also decided to add the ability to adapt the interface for the application being developed. There was Bootstrap for this.

Bootstrap is a free set of tools for creating websites and web applications. Includes HTML and CSS design templates for typography, web forms, buttons, labels, navigation blocks and other web interface components including JavaScript extensions.

Bootstrap uses modern developments in the field of CSS and HTML, so it is necessary to be careful supporting older browsers. [9]

**Interface design.** When developing the interface, a tab structure was used for ease of use – information is logically divided into tabs, which simplifies its search, does not clutter the main page and does not lead to the need to create additional forms and also reduces the time for loading other pages.

The main information about patients is stored in tables, over which it is possible to perform actions after selecting the appropriate data. This structure is the most simple and understandable for users with different levels of personal computer ownership. The use of standard controls allows you to minimize the learning time to work with the application and styling them with Bootstrap templates allows you to improve the appearance and attractiveness of this application.

The ability to download and view medical media data on the page was also used. This was done on the same pages where the patient data is selected. This approach is implemented using dynamic JSP data loading (figure 2).

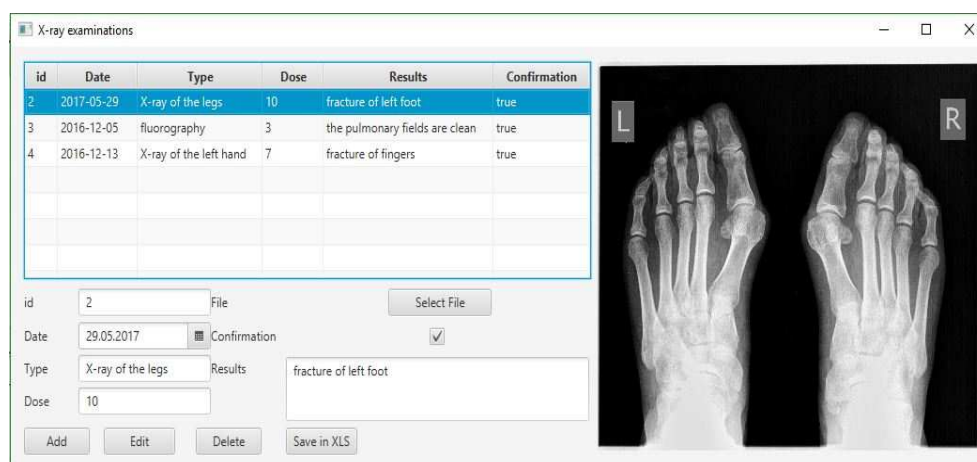


Figure 2. - An example of an interface with an image of the Blob type

## REFERENCES

1. Informatization of outpatient clinics [Electronic resource]. Access mode: [http://minzdrav.gov.by/ru/static/informatization/inform\\_apzveno](http://minzdrav.gov.by/ru/static/informatization/inform_apzveno). Mode of access: 20.01.2020.
2. Registration of a medical card of an outpatient [Electronic resource] - access Mode: [https://www.bsmu.by/downloads/kafedri/k\\_poli\\_ter/stud/v1.pdf](https://www.bsmu.by/downloads/kafedri/k_poli_ter/stud/v1.pdf) . Mode of access: 24.02.2020.
3. Model-View-Controller [Electronic resource]:<https://ru.wikipedia.org/wiki/Model-View-Controller>. Mode of access: 24.02.2020.
4. WebRTC API [Electronic resource]: [https://developer.mozilla.org/ru/docs/Web/API/WebRTC\\_API](https://developer.mozilla.org/ru/docs/Web/API/WebRTC_API) . Mode of access: 24.02.2020.
5. HTTPS Protocol - what is it? [Electronic resource]: <http://fb.ru/article/221368/protokol-https---chto-takoe>. Mode of access: 24.02.2020.
6. SRTP — what is a secure real-time data transfer Protocol? [Electronic resource]: <https://www.3cx.ru/webrtc/srtp/>. Mode of access: 24.02.2020.
7. Material from Wikipedia, a free encyclopedia. Java Server Pages [Electronic resource]: <https://ru.wikipedia.org/wiki/JSP> Mode of access: 24.02.2020.
8. Material from Wikipedia, a free encyclopedia. JavaServer\_Faces [Electronic resource]. Mode of access: [https://ru.wikipedia.org/wiki/JavaServer\\_Faces](https://ru.wikipedia.org/wiki/JavaServer_Faces) . Mode of access: 24.02.2020.
9. Material from Wikipedia, a free encyclopedia. Bootstrap [Electronic resource]. Mode access: [https://ru.wikipedia.org/wiki/Bootstrap\\_\(framework\)](https://ru.wikipedia.org/wiki/Bootstrap_(framework)). Mode of access: 24.02.2020.