Research and Development of the API for Personal Health Record

Anzhelika Parkhomenko[1] [0000-0002-6008-1610], Ivan Tyshchenko[2]
[1] Zaporizhzhia National Technical University, Zaporizhzhia, Ukraine
parhom@zntu.edu.ua
[2] Vanderslab Ltd, Zaporizhzhia, Ukraine
johan.willfred@gmail.com

Abstract. The problems of electronic personal health record development and application are considered in this paper. The features of popular on the market software for working with patients` medical information have been analyzed. It is shown that the main restrictions on the usage of the considered solutions in Ukraine are: the orientation towards integration with the state medical systems of another countries, the absence of localizations, the limited functions of searching the necessary data in previously downloaded information, closed API and the lack of documentation. Therefore, this work is devoted to the research and development of public API for Personal health record as well as its documentation based on the OpenAPI specification.

Keywords: informatization of medicine, personal health record, application programming interface, web-application.

1 Introduction

Thanks to modern information technologies, medicine acquires completely new features today. The widespread introduction of computer technology and specialized software leads to significant changes in medical theory and practice, as well as in the training of healthcare professionals. [1-3].

Today, a significant number of medical institutions of various forms of ownership provides Ukrainians with a wide range of services, especially in large cities. Due to the fact that the majority of private medical institutions are relatively small, they are usually specialized in certain healthcare sectors, that generally leads to the usage of several unrelated clinics by citizens. As a result, patients have medical documents (advisory opinions, research results, prescriptions, etc.) in various formats and forms. Therefore, the patient's medical information is kept distributed and most often inaccessible to the person himself. This makes difficulty in changing the place of medical services receiving and increases the number of examinations that have to be repeated. The lack of a holistic "history" of the patient increases the risk of incorrect diagnoses and reduces the effectiveness of the treatment as a whole.

All this factors contribute to the direct interest of people in the availability of their own medical information, regardless of the medical institution. Especially the collection and processing of medical information is complicated for people who travel

frequently or change their place of residence, which is quite common in the modern world. Thus, a personalized electronic medical record can provide the patients with the possibility to more closely monitor their health status and have access to the necessary health information at anytime, anywhere, from any platform. Moreover, it can be not only a medical history or medical examination data, but various information related to human health (physiological parameters and features of a healthy person's development, dietology issues, healthy lifestyle, medical bills, etc.) [4-7].

Today, a significant number of software systems are offered on the market for creating a personalized electronic medical record and working with it. Each solution has its own characteristics of implementation and usage. The selecting of an appropriate software is a complex task, because it requires to take into account the peculiarities of the process of informatization of medicine in Ukraine, the issues of personal data protection and reliable information storage [8].

Therefore, this work is devoted to the study of methods and ways of organizing personalized electronic medical records that will ensure the effective storage and retrieval of information about patient's health.

2 State-of-the-Art

As studies have shown, in [9] several important definitions are given related to the electronic information about the patients' health:

- Electronic Medical Record (EMR) is the electronic information related to the health of the patient, which is created, stored, maintained and used by certified medical professionals and staff in one medical institution.
- Electronic Health Record (EHR) is the electronic information related to the health of the patient, complying with national interoperability standards, which is created, maintained and used by certified medical professionals and staff of more than one medical institution.
- Personal Health Record (PHR) is the electronic information related to the health of the patient, complying with national interoperability standards, obtained from various sources, which the patient self-manages, controls and provides access to it [9].

Thus, EMR, EHR and PHR are the sets of information about the health of a particular person, differing from each other in terms of the collection of information and the way it is managed. These three concepts indirectly determine the most important areas of health care system informatization:

- Informatization of specific medical organizations.
- Creation of integration projects for the information exchange between medical organizations, at the national and international levels, that means the creation of a single information space for physicians (professionals).
- Creation of a common information space for non-professionals (patients and even those who do not consider themselves to be patients yet), that means the creation of services and facilities for maintaining medical information and electronic interaction between doctors and patients.

Informatization of medicine in Ukraine is an important part of the health care system reforming [10,11]. But, as studies have shown, today in Ukraine there are only attempts to create systems for working with EMR and EHR, with a significant lag of the third component - PHR. Considering the experience of the United States, Great Britain and other leading countries in which PHRs are actively used, their development and implementation is also a key step in the informatization of medicine in Ukraine.

The analysis of existing on the market software solutions for the PHR allowed to identify the most popular: Microsoft HealthVault, Zweena, Health Companion, Healthspek.

Microsoft HealthVault is a popular environment for working with PHR, which allows to collect, store, use and monitor information about the health of a person and his family, including medical images. This solution can be called one of the best on the market due to the well-known name of the developer - Microsoft company and a fairly large number of platforms on which different versions of HealthVault can be installed [12].

The studies have shown that the disadvantages of HealthVault are:

- Glut of functionality, for example, the program combines task manager and calendar functionality that it is usually more convenient for users to have them as separate standard applications for their platforms (for example, Google Calendar for Android smartphones).
- Ultimative approach to interface organization.
- Commercial overtones: the application of this software leads to the usage of the services of certain companies and promotes certain products that are often provided by Microsoft partners.
- Restrictions by regional principle: not all countries are available when creating a profile (for example, Ukraine is not available for selection).
- Lack of speech recognition and optical character recognition (OCR).

Zweena is an environment for working with PHR that supports import from HealthVault and requires an account and pre-registration to log in. This product has less functionality than HealthVault, and it can be used both for free and with a monthly payment [13]. Zweena is only available as a website and does not have its own mobile applications for working with it. The shortcomings that were identified during analysis of this software are:

- Completely non-adaptive interface, the web site is in no way adapted for use on a device with the screen size less than 960px.
- Lack of mobile applications.
- Closed API (information about it can be obtained only after contacting the company on the phone).
- Restrictions of free account (1000 entries maximum).
- Chargeable recognition of scanned documents.

Health Companion is a complete and long-term system for working with PHR that can integrate personal health information from various sources, track medical financial bills and make preventive health and wellness recommendations based on personal risk factors [14].

Health Companion has an adaptive web-site. Android and iOS apps are available for usage free of charge. The interface languages are English or Spanish, so the applications are oriented for usage in Americas. The system has a lot of integration with polyclinics and other medical services, and it also meets the requirements of the USA law.

The shortcomings of this system revealed during the analysis are:

- Orientation to receive information primarily through integration with other systems.
- Mobile applications have some issues with compatibility with the latest operating systems of the platforms which they operate on.
- Closed API.
- Limited search capabilities.

Healthspek is a free system for managing all personal and family health information in one secure location accessible from multiple devices anywhere in the world. The system has an adaptive website, Android and iOS applications that are available for usage for free. Interface language is only English [15]. The shortcomings identified during the analysis of this system are:

- Closed API.
- Adaptation problems for countries without insurance medicine.
- Limited information retrieval capabilities.
- Poor support for people with disabilities.

Summing up the results of analysis of the functionality and features of popular all over the world software systems for working with PHR, it can be concluded that most of them are aimed at integration with certain information providers, as well as the usage for certain countries. Another important limitation of the considered PHRs is the lack of documents` recognition as well as full search tools (only in HealthVault there is a relatively comprehensive search, but it also does not allow finding audio recordings by content). One of the major drawbacks of all considered software systems is the closed API. Nevertheless, API is an interface that allows developers to use ready-made blocks to build a program code and, thus, to speed up software development and make it more attractive economically [16-18].

Therefore, the task of development of an open API, which will enable the users to maintain their own electronic PHR, that will combine information from different sources and provide convenient tools for working with personal data, is relevant.

3 Architectural Approach and the Methods of Organizing the API for PHR

The REST implementation was chosen as an architectural style for the API-oriented development of PHR, which has several advantages over the SOAP approach [19-21].

The developed public API for implementing client applications (mobile, desktop and web) will provide basic PHR functionality: storing, viewing, and searching of the health information. In this case, the following possibilities are given to user:

- Creation of profiles for other people (family members).
- Creation of events.

- Storing of information in the form of photos, text and audio records.
- Search among stored information.
- Deletion of the records.
- Export of documents as PDF files.

Based on the completed assessment of the level of technologies and platforms development, it was decided to develop the system using the JavaScript programming language, which provides sufficiently high speed, convenience and performance of writing code. Node.js was used for the software interface on the server side development. During the analysis of the development toolkit, it was decided to use a number of libraries, such as Elasticsearch, Express.js, Mongoose, etc. These solutions make it possible to reduce the cost of program developing while increasing its quality parameters, and therefore to increase its competitiveness.

The Express was chosen as the Node.js framework, which provides easy maintenance, high speed, relatively cheap hosting and a wide range of hosting selection. Node.js support is the minimum for most hosting sites, and therefore there is no need to pay for “exoticism”. The flexibility of the structure and the minimum amount of time spent on performing routine actions are also typical for it.

WebStorm from etBrians was chosen as the development environment for the website. This environment is characterized by convenience and wide functionality. Its usage reduces time costs and increases programmer productivity. Developed API can be run on any modern Linux based VPS (Virtual Dedicated Server) server.

The created project can be divided into the following parts:

- ElasticSearch is the search engine, which provides high search performance among users` artifacts in the database.
- API documentation that provides developers with information on incoming and outgoing data for the route.
- API-core is the part of the system that is responsible for user requests routing and processing.

The developed API is the application that installs on an ordinary web server with Node.js 6 (LTS Boron), MongoDB 3.4. *, Elasticsearch 5.6. *. It is available for the usage by clients via the http / https protocol. JSON data exchange format is used.

It is necessary to perform the following steps for developed API functions usage.

Step 1. To form the HTTP request that includes the following data: server address, the API version (/ api / 1), the controller name (/ users), the action (/ profile). Parameters can be transmitted both via the URL "? Param = 123" and in the body of the request.

Step 2. To send the generated request to the server (at the specified address) via the GET / POST / PUT / DELETE method: POST https://server-url.com/api/profiles

Step 3. In response, to get the result of the command execution by the server in the form of JSON:

```
{
  "fullName": "string",
  "phone": "string",
  "birthday": "string",
  "relationship": "string",
```

```
    "gender": "string",
    "height": {
      "ft": 0,
      "in": 0
    },
    "weight": "string",
    "emergencyContactName": "string"
}
```


The created list of requests to API is as following.

POST / auth / sign-up – the user registration.

POST / auth / sign-in – the user authorization.

POST / auth / forgot – the password recovery.

POST / profiles / - to create a profile.

PUT / profiles / {id} – to edit a profile.

DELETE / profiles / {id} – to delete a profile.

POST / profiles / {profile} / events – to create a medical event.

GET / profiles / {profile} / events – to get a list of profile medical events.

DELETE / profiles / {profile} / events / {id} – to delete the event.

GET / profiles / {profile} / events / {id} – to get information about the event.

PUT / profiles / {profile} / events / {id} – to edit the event.

POST / profiles / {profileId} / events / {eventId} / recordings – to add an audio record;

DELETE / profiles / {profileId} / events / {eventId} / recordings / {recordId} – to delete an audio record;

GET / aprofiles / {profileId} events / search – the search of information.

GET / profiles / {profileId} / events / {eventId} / notes - the list of notes.

PUT / profiles / {profileId} / events / {eventId} / notes – to add the notes.

DELETE / profiles / {profileId} / events / {eventId} / notes – to remove the notes.

POST / profiles / {profileId} / events / {eventId} / photos – to upload photos.

DELETE / profiles / {profileId} / events / {eventId} / photos / {id} – to delete photos.

The convenience of working with any API substantially depends on how its documentation is written and executed.

There are different specialized services and tools (API BluePrint, Swagger, API Designer, RESTUnited, MireDot, Kittn API, apiDOC, apiGen and others) for API documentation development. All generators have their own advantages and disadvantages. Some of them are not for free and support limited number of programming languages. The process of documents generation is based on the usage of standardized format (Markdown, JSON, YAML, RAML). For example, the API BluePrint uses the Markdawn format which is intended primarily for text formatting and not for the usage as a basis of documentation generation. The Swagger supports YAML and JSON formats. The YAML is much more convenient than the JSON, but it does not allow to describe the repeating elements easily, which are often present in the API descriptions. So, it is very difficult to adjust it under API documenting. API Designer is the platform for testing as well as the interactive editor, based on RAML

format for creating documents online. The simplicity and consistency are its advantages, but the lack of tools developed by the community is its open issue [18].

So, the Swagger was chosen for API documentation organization and management after the provided analysis [22]. Swagger User Interface is a small collection of scripts for creating interactive documentation for APIs of web applications with the REST architecture.

The main Swagger online documentation page is shown in Fig. 1. There are two text fields on the top panel. The first field contains the path to the JSON documentation file. The developed API methods are described in this format. The second field is the access token that will be used for requests to the API creation with the usage of Swagger. During the page opening, the value of the token is taken from the configuration file and the test user is added to the database using migration. If it is necessary, the access token can be changed with using the documentation. For example, for several users work simulation in the application, it is necessary to find the access token of the desired user in the database. Then enter this token in the field and click the Explore button. The token will be valid until the user exits this page. The token from the configuration file will be loaded after the page re-opening.

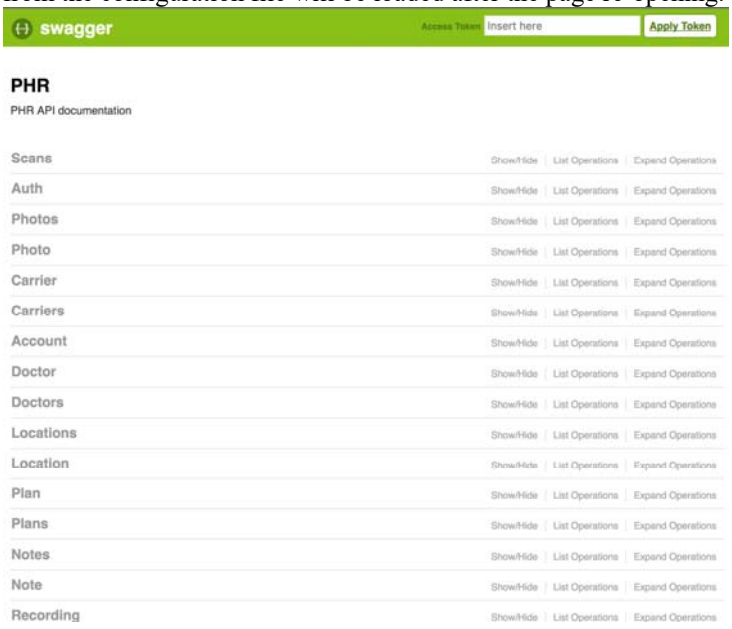

Fig. 1. API documentation page.

The documentation page contains the accordion interface element with addresses which are contained in the API. A brief information about the API (the base URL that is attached to the server address and the current version of the API) is shown after the list of display controllers. The list of groups` routes is opened with clicking on its name (Fig. 2).

It is possible to find out from the list next data: the method which you can call this route with (POST / GET / ...), the required URL that is attached to the base and a brief description of the assignment. Routes are sorted by call method and alphabetical order. If the method is selected, detailed information about it opens: a text description, a list of parameters and responses from the server, as well as a “Try it out” button for creating a request to the server (Fig. 3). The result of calling the API method is displayed after clicking on the button (Fig. 4).

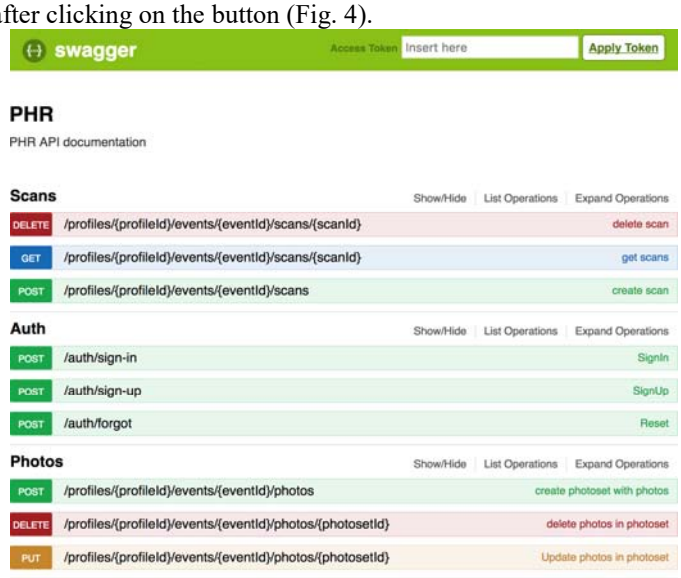

Fig. 2. Grouped API resources

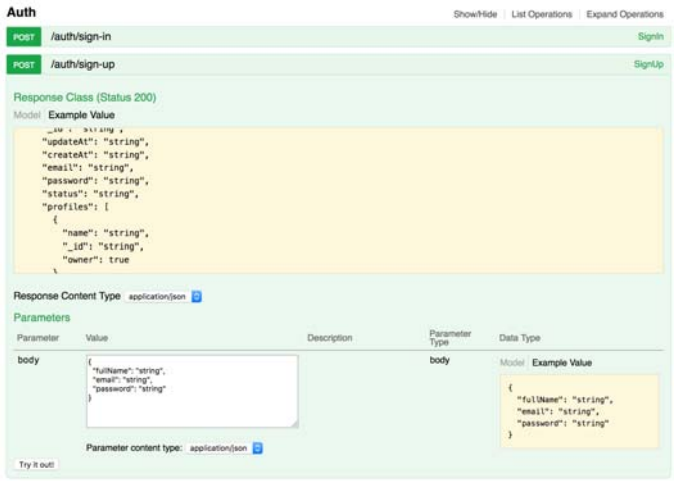

Fig. 3. Description of the method

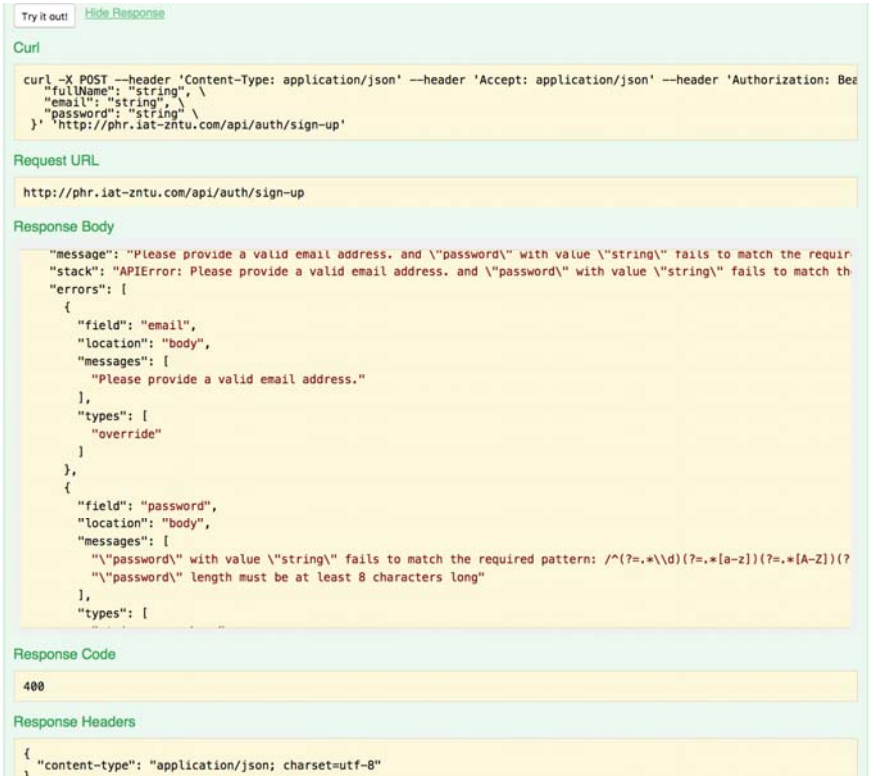

Fig. 4. Answer from the server

4 Conclusion

Nowadays Ukraine is only at the beginning of informatization process of medicine. Considering the experience of the United States, Great Britain and other advanced countries, the implementation of PHR is a relevant task and it can be considered as an important step in the informatization of medicine, together with the introduction of EMR and EHR. Therefore, the problems considered in this paper are relevant, and the proposed solution is seemed to be promising. As a work result, an open API was created, which simplifies the development of mobile and web applications for the necessary platform for working with PHR, and it also provides visual documentation according to the OpenAPI specifications.

Acknowledgements. This work is carried out partly with the support of Erasmus + KA2 project BIOART «Innovative Multidisciplinary Curriculum in Artificial Implants for Bio-Engineering BSc/MSc Degrees» (586114 – EPP – 1 – 2017 – 1 – ES - EPPKA2 – CBHE – JP)

References

1. Kachmar, V.O.: Medical information systems - the state of development in Ukraine. Ukrainian Journal of Telemedicine and Medical Telematics, **8** (1), 12-17 (2010)
2. Churpiy, I.K., Churpiy, N.V., Skripko, V.D. Current state of informatization in medicine. Bukovinsky medical bulletin, **15**(1), 171-173 (2011)
3. Nazarenko, G.I., Guliyev, Ya.I., Ermakov, D.E.: Medical information systems: theory and practice, general editor G.I. Nazarenko, 320 p. FIZMAT-LIT, Moscow (2005)
4. Bates, D.W.: Physicians and ambulatory electronic health records Health Aff, Millwood, **24** (5), 1180-1189 (2005)
5. Menachemi, N., Perkins, R.M., Durme, D.J.: Examining the adoption of electronic health records and personal digital assistants by family physicians in Florida. Inform. Prim. Care, **14**(1), 1-9 (2006)
6. Miller, R.H., Sim, I.: Physicians' use of electronic medical records: barriers and solutions. Health Aff, Millwood, **23**(2), 116-126 (2004)
7. Mark, A., Hall, J.D., Kevin, A., Schulman, M.D.: Ownership of Medical Information, JAMA, **301**(12) 1283 (2009)
8. Gravanova, Yu.: Features of the storage of medical information. http://www.cnews.ru/reviews/free/national2006/articles/safe/
9. Defining Key Health Information Technology Terms. http://www.hitechanswers.net/wp-content/uploads/2013/05/NAHIT-Definitions2008.pdf
10. Decree of the President of Ukraine No. 187/2012 The National plan of actions for 2012 on the implementation of the program of economic reforms for 2010-2014. "Prosperous Society, Competitive Economy, Efficient State". https://zakon.rada.gov.ua/laws/show/187/2012
11. Draft Order of the Ministry of Health of Ukraine "On approval of the concept of informatization of the healthcare of Ukraine on 2013-2018 years" http://uacm.kharkov.ua/download/2013_10/148-154_Konzepziya_10_sc_P.pdf
12. Healthvault. https://international.healthvault.com
13. Zweena Health. https://www.pinterest.com/zweenahealth/
14. Health Companion. https://www.healthcompanion.com
15. Healthspek. https://www.healthspek.com
16. Anuff, Ed.: APIs are Different then Integration. https://pages.apigee.com/rs/apigee/images/APIs-not-integration-ebook-05-2014.pdf
17. Poliakov, M., Larionova, T., Tabunshchyk, G., Parkhomenko, A., Henke, K.: Remote laboratory for teaching of control systems design as an integrated system. In: The XIII international conference on Remote engineering and virtual instrumentation, Madrid, Spain, pp. 333-340 (2016)
18. Parkhomenko, A., Gladkova, O., Sokolyanskii, A., Shepelenko, V., Zalyubovskiy, Y.: Implementation of reusable solutions for remote laboratory development. International Journal of Online Engineering. **12**(7), 24–29 (2016)
19. Why Should We Choose REST (Client-Server) Model to Develop Web Apps? https://medium.com/@audira98/why-should-we-choose-rest-client-server-model-to-develop-web-apps-c3bb2451b13a
20. Types of HTTP requests and REST philosophy. http://habrahabr.ru/post/50147/
21. W3C. SOAP Specification. https://www.w3.org/TR/soap/
22. Maltseva, D. Why use Swagger for creating and documenting APIs. https://dev.to/dianamaltseva8/why-use-swagger-for-creating-and-documenting-apis-115l