

A Cartography and Information System of Hydrochemical Data

Kristina O. Beym, Vladislav S. Sherstnev, and Dina F. Mymrina

Abstract — The paper focuses on the process of engineering and developing a cartography and information system dealing with the analysis of hydrochemical data aiming at monitoring conditions of surface water bodies. The problem-solving procedure is discussed, with its benefits and weaknesses being analyzed. The system under consideration is based on the data on Siberian Federal District. It uses modern information technologies such as web-access, database management systems (DBMS), service-oriented architecture, asynchronous Javascript technology and XML (Ajax), Leaflet (JavaScript library for mobile friendly interactive maps), and some others. The developed system appears to be a convenient way to provide specialists, dealing with the evaluation of potable groundwater quality, with the data on surface water bodies.

Index Terms— Hydrochemical data, service-oriented architecture, thematic mapping, monitoring.

I. INTRODUCTION

In recent years the resources of the Russian ground potable water have been estimated, which allows setting the most urgent tasks and the main directions in the development of mineral resource base of potable water and its use. One of the critical tasks appears to be the creation of the bank of groundwater quality research data. This task is essential to improve the water supply in the regions where high content of chemical macro- and micro-components in potable groundwater was discovered [1]. The research is due to a lack of uniform methodological approach and a universal hydrogeochemical map (at a scale of 1:1 000 000). This results in the fact that most of the findings and created maps do not correlate with each other within the unified hydrogeological and hydrochemical model. So, it appears that one way of solving this problem could be the design and development of a cartography and information system, which stores hydrochemical data on hydro-objects in the Siberian Federal District.

While creating the system, the following tasks should be fulfilled:

- examination and analysis of the subject domain; design and development of a hydrochemical database;
- selection of a set of software tools and technologies;

Manuscript received December 23, 2015; revised January 19, 2016.

Kristina O. Beym, National Research Tomsk Polytechnic University, Tomsk, Russia (corresponding author to provide phone: +7-906-198-8222; e-mail: kob4@tpu.ru).

Vladislav S. Sherstnev, National Research Tomsk Polytechnic University, Tomsk, Russia (corresponding author to provide phone: +7-913-889-7557; e-mail: vss@tpu.ru).

Dina F. Mymrina, National Research Tomsk Polytechnic University, Tomsk, Russia (corresponding author to provide phone: +7-913-889-7557; e-mail: mymrina_df@tpu.ru).

- development of a set of required functions;
- design and development of a graphical user interface (GUI) of the web-application.

II. VARIANTS OF SOLUTIONS AND THEIR ANALYSIS

The development of a special cartography and information system can be based on different software platforms.

One of the variants is to use a full-function geographical information system (GIS) as a platform where all hydrochemical data can be loaded and stored. In this case, it may require developing relevant modules for certain GIS which could adapt it for processing hydrochemical data. The use of a full-function GIS such as ArcGIS [2] or MapInfo [3] would not be possible without purchasing licenses. Thus, it appears to be a more expensive approach than using free software. Besides, the use of universal full-function GIS requires additional maintenance charges, operation cost, staff training costs, etc. In addition, the task in hand does not require as many functions as a modern full-function GIS can offer. For this reason, the idea to use a full-function GIS was dismissed.

The second suggestion on how the system could be developed is to create a special desktop application, which uses GIS-components in its core to map the data. The necessity to use licenses for GIS-components and aging of desktop applications are the drawbacks of this variant.

The authors of the paper have chosen the third variant, allowing the development of a web-based cartography and information system.

The interaction between the user's web-browser and the web-application is via the HTTP protocol. The web-application is responsible for the creation of the graphical user interface (GUI) of the system and substrate mapping, which is due to interaction with known suppliers of free map data [4], [5], [6]. At the same time, the web-application is not involved in any data processing. All the requests for processing data are transmitted from the web-application via SOAP Protocol [7] to the specialized web-service that implements all the business logic of the system. The web-service has a wide range of methods that can solve some of the user tasks:

- provide numeric data on hydrochemical objects;
- form a thematic layer which could allow mapping web-applications, etc.

In its turn, the web-service accesses data for its calculations in a separate database that stores this information.

The advantages of this approach are as follows:

- the web-based information system is platform-independent for end-users [8];
- in contrast to the above discussed variants, the web-application does not require installation of any additional software in order to work with cartographic information;
- due to the use of web technologies, one can work with the application from virtually any location where there is an access to the Internet.

III. DESIGN

Design is one of the most important and crucial steps in the implementation of any project. The quality of the system created within the project determines whether it will be necessary to review the basic concepts of the information system in future and make fundamental changes in it, which is always a time- and resource-consuming task.

A. Architecture Design

When designing the system architecture, its main purpose was seen as the visualization of the results of the hydrochemical research on the map. At the same time, the architecture was designed to allow further increasing the functionality of the system. The “Deployment diagram” which is modelling the physical deployment of artifacts on the nodes of the system is shown in Fig. 1.

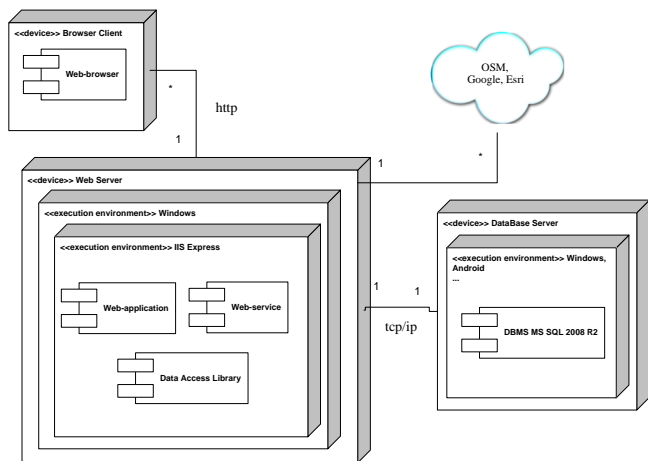


Fig. 1. Deployment diagram.

As can be seen from Fig. 1, the web-application and the web-service are located on a web-server, but, in fact, they can run on different servers. As a provider (supplier) of external map data different Internet services may be used (e.g. OpenStreetMap, GoogleMap, Yandex, Bing and others). Part of “use cases” diagram is shown in Fig. 2.

Some of the important tasks of the system include:

- display of database which accumulates hydrochemical data (obtained from tests) on water objects in the form of charts;
- management of layers containing different geological and hydrological objects;
- provision of basic mapping tools (navigation, scale).

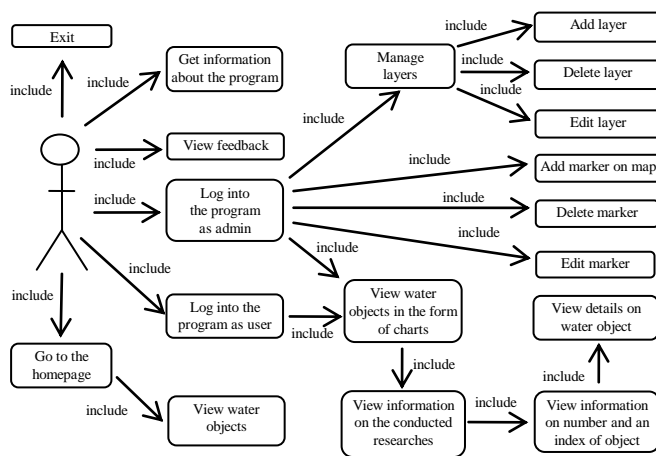


Fig. 2. “Use cases” diagram.

B. The Design of the Database Structure

The design of hydrochemical database was based on the results of the analysis of the data which were obtained from the regional center of the Subsoil State Monitoring and the information-analytical system for groundwater state monitoring (developed by the State Scientific “VNIIGeosystem” [9]). All the source data mentioned above were obtained in an Excel-file format and required restructuring.

The analysis of the material obtained has allowed developing a conceptual model and then a physical data model to store hydrochemical information. The database structure design was performed with the help of Oracle Toad Data Modeler [10], a database design tool. Part of the physical database model is shown in Fig. 3.

When designing, the rights to access the entities of the database (when performing standard “create, read, update, delete” (CRUD) operations) were defined.

The main data are stored in the six tables of database management system (DBMS).

The table “Users” is used to store usernames of the registered users, their roles and passwords. The passwords are encrypted using an encryption algorithm “MD5”.

The table “GeoData” stores the data on the spatial location of hydro-objects (the serial number of an object in the catalogue, latitude, longitude).

The table “AttrData” contains a large amount of attribute information on the facts of the hydrochemical research:

- the number and depth of the hydro-object;
- the type of geological systems;
- information on quantitative and qualitative composition of mineral and organic compounds;
- the date of the first and the last tests;
- the number of tests with micro-components, nitrogen and organic compounds; radiation parameters; macro-components and overall performance of the geochemical status of groundwater.

The table “IndexData” was designed to store information on the index of the aquifer (water-bearing stratum).

The table “GeoSystems” contains information on the geological system of water bodies. Also, it stores information on the layers and maps names that refer to this system.

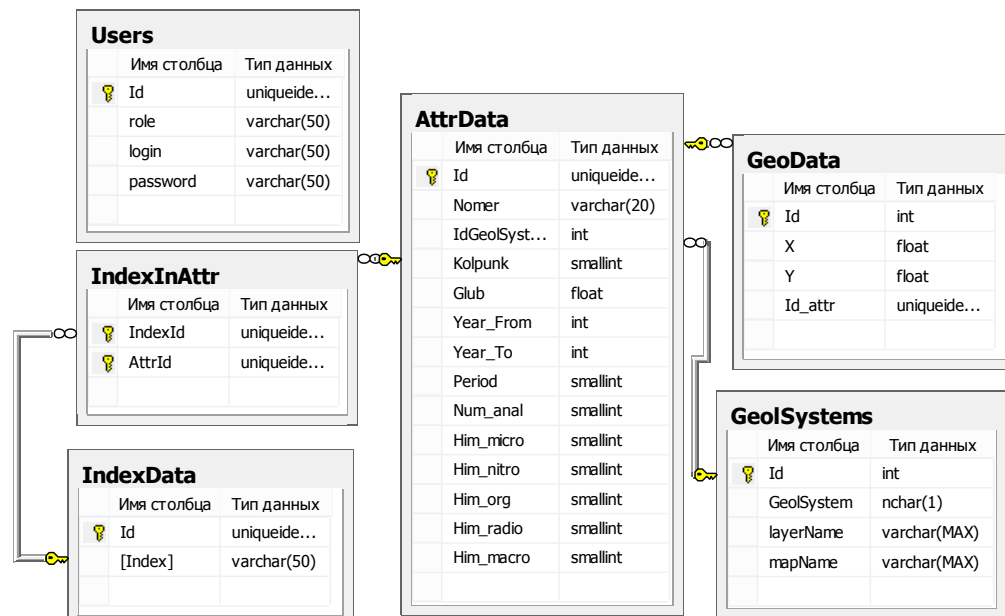


Fig. 3. Part of the physical database model.

C. The Design of Classes

The design of the system on the whole was based on the service-oriented architecture. EDM ADO.NET model was used to access the data in the database. For this reason, classes shown in Fig. 4 were designed.

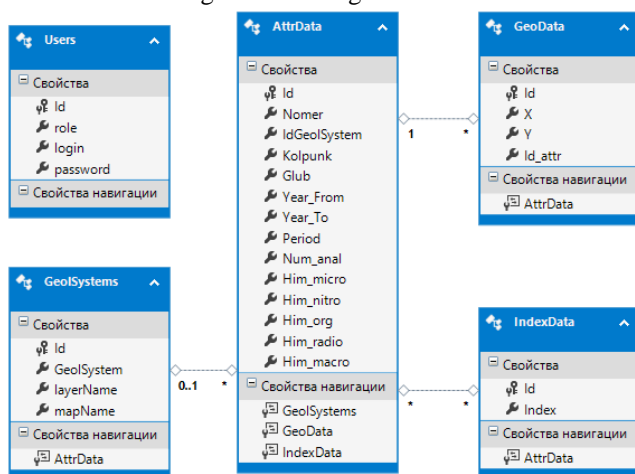


Fig. 4. Classes of EDMX-model.

As shown in Fig.5, the basic structure of the web-application consists of five controllers:

- the “Account”, whose function is to implement authorization logic;
- the “Home”, whose function is to manage data, available by unauthorized users;
- the “User”, available to authorized users (as a “user” or the “admin”);
- the “LayerController” - ApiController, which is responsible for adding, deleting and editing data layers;
- the “ObjectController” - ApiController, which is responsible for adding, deleting and editing data on objects.

Access to an action of a particular controller is limited by the role of the user, connected to the system. There are three user roles in the system:

- an unauthorized user (role “guest”);
- a normal (authorized) user (role “user”);

- the system administrator (role “admin”).

The unauthorized user has rights to view only descriptive information on the system, without access to the actual hydrochemical data.

A normal (authorized) user can read the data on hydro-objects that are stored in the system and view these data on the geographic maps.

The system administrator (“admin”) has most of the opportunities to manage the system: they can add data to the map and database, edit and delete them. Controllers cause “view”-components to display data.

The controller “Home” contains three so-called “full”-views: “Index”, “About”, “Contact”. These views display such basic information on the application as: information on the main page (“Index”), the summary of the features of the software (“About”), feedback information (“Contact”).

It is these views that are available for an unauthorized user of the system with the “guest”-role. In addition, to visualize the map on the main (“Home”) page the so-called “partial” view “MapIndex” was created.

The “Account” controller contains the “Login” view, which is necessary for user authentication. The “User” controller contains two “full” views: “AdminMain”, which is available to the administrator of the system; “UserMain”, which contains information available to an authorized user of the system. To visualize the card in the “ful” views, an additional “partial” view “mapN” was developed. To display information on the selected marker (water-bearing object) the “partial” view “MoreInform” was implemented.

The “Object” controller contains actions for “Get”, “Post”, “Put” and “Delete” procedures, which allows obtaining all information on hydro-objects, adding hydro-objects, changing any of the information on them, and removing markers of hydro-objects from the map. The controller “Layer” provides similar actions “Get”, “Post”, “Put” and “Delete” to manage the set of layers on the geographic map.

Access to the data of API-controllers is via JavaScript-code from “views”.

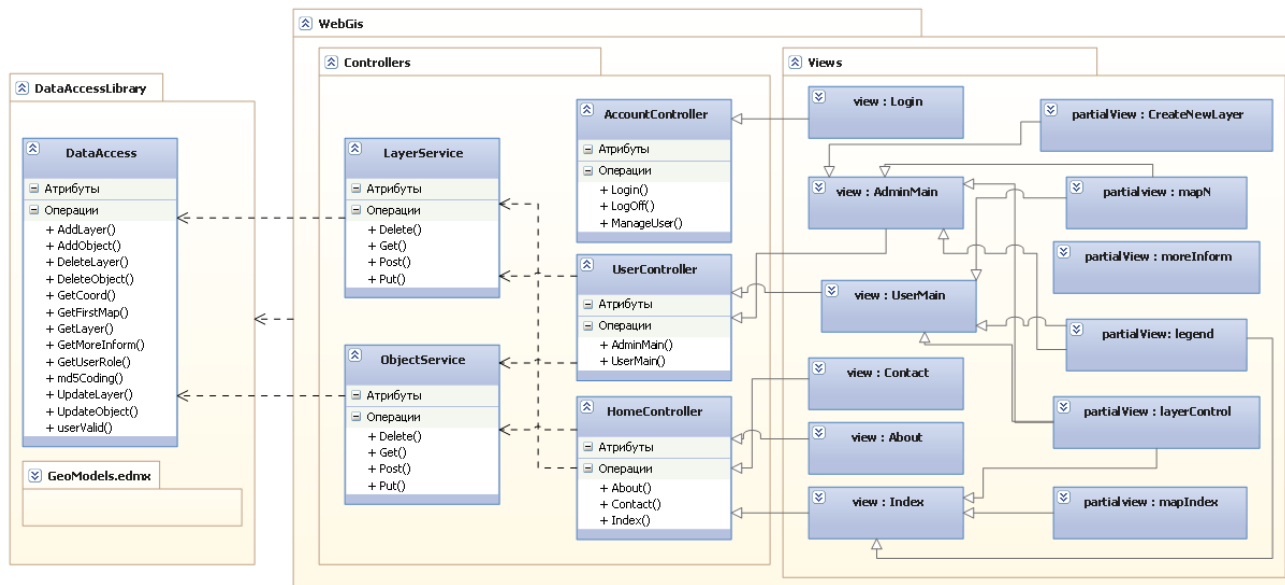


Fig. 5. The classes structure of the system.

Besides, additional templates of views were created to be used in all the views: “_Layout”, a common template for all views; “_Layers Control”, a common template which allows visualizing the layer list and the logic of their management.

Layers are a group of objects that are put on the map at a time. Only one specific layer can be visualized on the map at a particular time. “_LoginPartial” is an element which visualizes the top-header part of web-pages.

Here the following elements could be shown: the organization’s logo, buttons to escape to other views (e.g. an authorization view or exit from the current session). The Ajax-technology (Asynchronous Javascript and XML) [12] was applied to build the web-application interface. Due to the Ajax-technology, when you update your Web-page, it does not fully reload, and the Ajax updates only the desired part of the content of the web-page. As a result, the speed of the web-application increases. Using the Ajax technology, we can display on the pages the partial “view” of the map, which contains the details of the selected hydro-object. Fig. 5 shows the basic parts of the system which describe the designed structure.

IV. THE OPTIMIZATION OF ALGORITHMS SYSTEM

In the process of designing algorithms of the system, it was discovered that downloading and visualizing multiple data on hydro-object research findings on the map should occur simultaneously. In terms of the user interface, the display of all the stored data on a single map appears unreasonable. Obviously, such an operation will lead to the simultaneous transmission of a large amount of information from the database to the user via web-services and web-application. At the same time, the end-user will not see the correct display of the data, as many of the data will be displayed on the map, overlapping each other. To mitigate these effects and to optimize the interaction between the parts of the system (DBMS, web-service, web-application), the authors suggest using appropriate algorithms for the generalization of spatial data (points and graphs illustrating the results of hydrochemical research). One way to optimize

the transmission and display of spatial data (used to solve the problem of generalization) is a “window spatial queries” method [13].

To be able to use this method in the developed system, we had to modify the algorithm that selects the data on the location of hydrochemical objects from the database. At step 1, the “spatial query window” is sent from the client web-application to the web-service and the database.

The query contains the spatial coordinates of the viewing area on the map which is on the client web-application.

At step 2, the database performs the appropriate spatial query concerning the location of hydrochemical objects, choosing the data, whose coordinates are located inside the “spatial query window”, and then it prepares these data to the client web-application.

At step 3, DBMS excludes from the final set of the hydrochemical objects those objects that have been previously transmitted to the client web-application. Via the “spatial query window”, only the required objects (filled objects with solid border, as in Fig.6) will be transmitted to the client web-application.

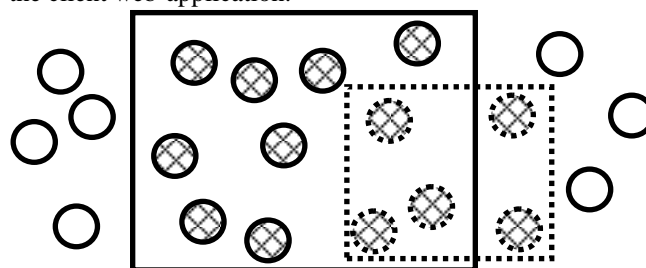


Fig. 6. “Spatial query window” example.

These are objects that are in the area of “spatial query window” (these are the objects within the rectangle with solid border, see Fig.6) which were not previously transmitted to the client web-application (filled objects with point border, as in Fig.6). The other objects (without any fill, see Fig.6) are not transmitted to the client web-application. Another way to optimize is to transfer only the data that the client will be able to see in the current resolution [14], [15]. Thus, the important steps here are:

- identification of the current scale of the map displayed on the client web-application;
- transfer of the scale value and the client's ID to the web-service and the database;
- identification (by the database spatial analysis function: search for the intersections with the buffer zones of the previously transferred objects with other objects) of the objects that should be transferred to the client web-application;
- exclusion from the list of those objects that are too close to each other (for a given value scale) (Fig.7,8).

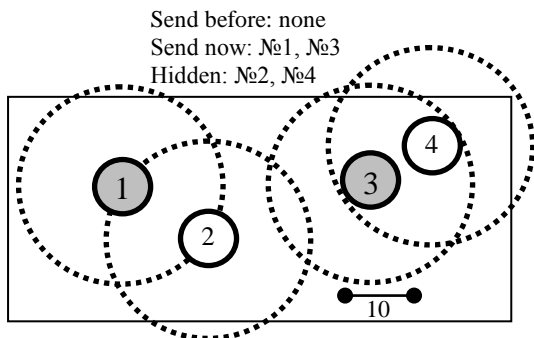


Fig. 7. The objects chosen for transfer before zoom (at a distance).

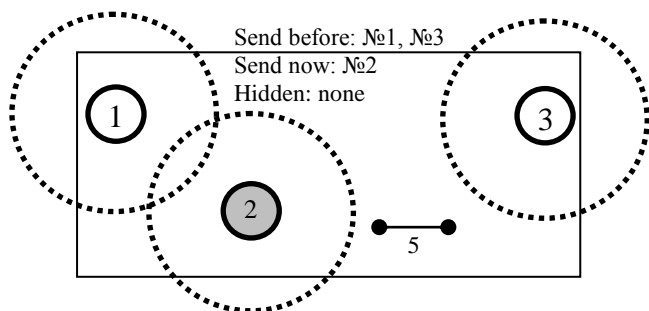


Fig. 8. The object chosen for transfer after zoom (at approach).

As a result, the client web-application receives and displays the required data in the shortest intervals of time. The algorithms described optimize the amount of transmitted spatial data and the speed of the response of the system on the whole, which is due to more complex algorithms used to process spatial data on the side of the web-service and the database. It should be noted that these algorithms are used exclusively to optimize the processing of object-oriented spatial data (the location of hydrochemical objects).

V. SOFTWARE COMPONENTS TO BE CHOSEN AND IMPLEMENTED IN THE SYSTEM.

To store the information it is necessary to select the database management system. Microsoft SQL Server's share of the market, according to Gartner, accounts for 46.8%, the other share of the market belongs to Oracle and IBM DB2. The DBMS such as Open Source, Postgre and Firebird have a relatively little market share. According to the report by Gartner, a leading market position among databases is occupied by Microsoft SQL Server. For this reason, MS SQL Server was chosen as DBMS.

An open source library "Leaflet" will be used to render and display geographic maps on the web page. The library was created using JavaScript-code in 2011 by Vladimir Agafonkin [16]. The main features of the library are: ability

to work in browsers, mobile devices, and desktop browsers; a small size of the library (33 kb); a well-documented API; functionality, which can be expanded by additional plug-ins.

The library is widely used on such sites as Flickr, Foursquare, Craigslist, Data.gov, IGN, Wikimedia projects, Meetup, OpenStreetMap, WSJ, MapBox, CloudMade, CartoDB, and others. The library can be downloaded from the official website.

VI. THE CHOICE OF DEVELOPING TECHNOLOGY

The ASP.NET technology, using C# programming language, was chosen as the language for the system development. There are two basic approaches to the creation of web-applications using C# code: "WEB-form" and "MVC" [11]. The former approach (WEB-form) uses the ASPX-template, where each online-resource (HTML-page) is a collection of ASPX-pages and the software (code) that controls the generation of the final HTML-page. The latter approach (MVC) is based on the template which is to separate the business logic and graphical user interface (GUI) in the applications. This approach allows developers to modify separate parts of the application without affecting the remaining parts. The MVC pattern was selected as a template to design the system architecture, given that it is more modern and allows modifying the system components, if necessary. Today, ADO.NET "Entity Framework" [17] and "Linq To Sql" [18] are the most commonly used programming techniques of the "object-relational mapping" (ORM) which work with ASP.NET. The main advantage of "Entity Framework" (EF) is that it is modern, powerful, and is recommended by the Microsoft data access technology. Using EF, developers do not need to write most of the code for data access, which is usually required in "Linq To Sql". Besides this, "Linq To Sql" uses the so-called "LazyLoading", which is the technology that secretly loads the associated data together with the requested data. In contrast, EF has no such "LazyLoading". Thus, the object-oriented technology ADO.NET "Entity Framework" will be used as a technology that allows access to data stored in the database.

VII. IMPLEMENTATION

The next step in the web-system development is the realization of the system. The developed web-system consists of a set of dynamic web-pages interacting with the web-service, DBMS and the provider of the cartographical information. The homepage is displayed at the start of the programme (Fig. 9).

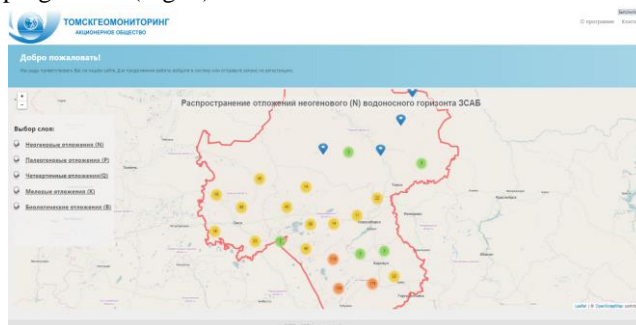


Fig. 9. The homepage of the client web-application.

The user can move to the pages, containing feedback data, information on the system, and the page for authorization. As the user is not authorized, they cannot see the exact information on the number of hydrochemical objects, but is able to see markers on the map (which are indicated by an icon).

The logic of the authentication and authorization is realized: the user moves to the page of authorization and enters the required data. Then, the authentication occurs in the application by matching the password entered and the user's account password stored in the database. After that, the user logs in and automatically goes to the page corresponding to his role. The normal user (who authorizes as a "user") has only one page containing a map (Fig. 10).

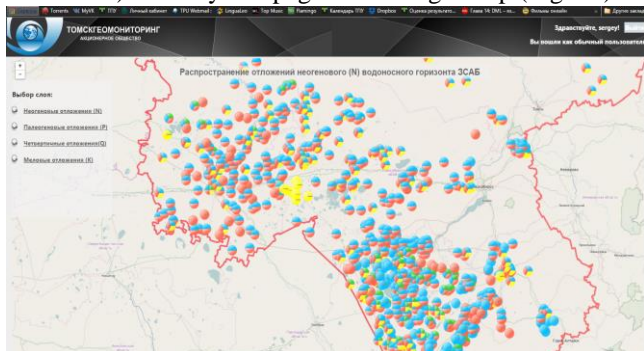


Fig. 10. The homepage of the normal user.

This page displays the data on water-bearing objects in the form of charts. The map is shown as a partial view "mapN", and when another layer is chosen, the map updates its objects using Ajax technology. When the user clicks on the marker, he views the summary about the groundwater object. This summary includes the index of the aquifer and the item number. Also, here the user can see the detailed information on the hydro-object (Fig. 11).

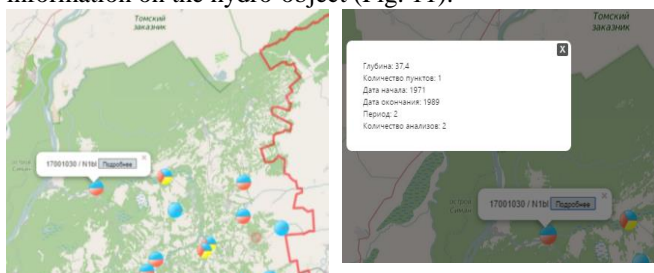


Fig. 11. The details of the hydro-object.

The system administrator after authorization appears on the "AdminMain" page, which is similar to the homepage of the normal user. At the same time, it contains some elements that can be used to change the information in the database.

VIII. TESTING

Functional testing at the system and component levels was held. Unit testing was chosen as a way to construct tests. "Unit" library was used for the purpose. The "xUnit" was used a tool for testing the system. The unit-test, aiming at testing methods, was created for each controller. Besides, the work of the system was tested by a number of users. The system correctly worked under operational loads; deadlocks did not occur. The analysis of the performance was also made: performance bottlenecks were revealed and

suggestions on how the problem can be solved were made.

According to the results of the testing, it was proved that the software application works properly and is ready for practical application.

IX. CONCLUSION

The result of the research is the development of the software which includes a web-application, managing hydrochemical data. This software can be used by those who are involved in the study of the groundwater qualitative composition. The work on the system will be continued; its improvement and realization of new features is planned.

The described way of developing the software aimed at monitoring hydrochemical data can be used in the development of similar systems dealing with the accumulation and processing of information on thematic objects.

ACKNOWLEDGMENT

The authors would like to thank NR TPU and JSC (joint-stock company) "Tomskgeomonitoring" for their assistance in this project.

REFERENCES

- [1] Official web site of JSC "Tomskgeomonitoring". [Online]. Available: <http://www.tgm.ru/>
- [2] ArcGis. ESRI. [Online]. Available: <https://www.arcgis.com>
- [3] MapInfo. Pitney Bowes Inc. [Online]. Available: <http://www.pitneybowes.com/us/location-intelligence/geographic-information-systems.html>
- [4] Bing. Microsoft Inc. [Online]. Available: <http://www.bing.com/mapspreview>
- [5] Yandex Maps. Yandex. [Online]. Available: <https://maps.yandex.ru>
- [6] Google Maps. Google Inc. [Online]. Available: <https://maps.google.com>
- [7] SOAP. World Wide Web Consortium. [Online]. Available: <http://www.w3.org/TR/soap/>
- [8] Web application. [Online]. Available: https://en.wikipedia.org/wiki/Web_application
- [9] Projects of Russian research institute of geological, geophysical and geochemical systems (Vniigeosistem). [Online]. Available: <http://www.geosys.ru/index.php/en/projectinform/74.html>
- [10] Toad for Oracle. Dell Inc. [Online]. Available: <http://software.dell.com/products/toad-for-oracle>
- [11] Model-view-controller. [Online]. Available: <https://en.wikipedia.org/wiki/Model-view-controller>
- [12] AJAX. [Online]. Available: [https://en.wikipedia.org/wiki/Ajax_\(programming\)](https://en.wikipedia.org/wiki/Ajax_(programming))
- [13] Querying Spatial Data. Oracle Corporation. [Online]. Available: https://docs.oracle.com/cd/A87860_01/doc/inter.817/a85337/sdo_objq.htm
- [14] The distributed geographic information system for collecting and processing of production information in the gas company. Sherstnev V. S. [Online]. Available: <http://www.lib.tpu.ru/fulltext/a/2005/45.pdf>
- [15] Hierarchical structures implementation methods on large databases. Sherstnev V.S. Proceedings - 9th Russian-Korean International Symposium on Science and Technology, Korus-2005. Novosibirsk State Technical University. Novosibirsk, 2005. p. 696-700.
- [16] Leaflet. Java Script library for mobile-friendly interactive maps. [Online]. Available: <http://leafletjs.com/>
- [17] Entity Framework. [Online]. Available: [https://msdn.microsoft.com/en-us/library/gg696172\(v=vs.103\).aspx](https://msdn.microsoft.com/en-us/library/gg696172(v=vs.103).aspx)
- [18] LINQ to SQL: .NET Language-Integrated Query for Relational Data. [Online]. Available: <https://msdn.microsoft.com/en-us/library/bb425822.aspx>