# Development of Client-Server Technology for Access to the Database of Algorithms on the Vue.Js Platform in Educational Process

Alexander Tarasyev[1], Victoria Turygina[1], Mark Pavlov [2], Yuriy Kharitonov[2] and Danil Soltys[1]

[1] Ural Federal University, 620002, 19 Mira street, Ekaterinburg, Russia
[2] Donetsk National Technical University, 83015, 58 Artema Street, Donetsk, Ukraine
v.f.volodina@urfu.ru

**Abstract.** In this paper, the creation of an information system is considered, the main task of which is to provide information on algorithms in a form accessible for training, namely a general description of the algorithm, examples of implementations in programming languages, links to sources with additional information, visualization of their work. In addition, there are tasks to develop such functions: comparing performance algorithms, step-by-step execution of the algorithm with reference to its visualization. The methodology for developing a client-server application is demonstrated, the architecture of the system as a whole is described, and its individual elements are described in detail: the client part is the Vue.js framework, as well as its add-ons: for rapid prototyping of the user interface (Vuetify), for managing the global state of the application (Vuex); the server part, which is a H2 relational database, as well as a server application written in the Java programming language using the Spring MVC framework, which involves the allocation of layers such as controllers responsible for redistributing requests from clients to services that have access to special repositories that , in turn, refer to the layer that implements the database interaction interface (DAO). The result was a prototype application in which the basic functions are implemented, divided into two roles: a regular user with the rights to view content, and an administrator who has the ability to both view materials, and create, edit and delete them; as well as directions for the further development of this information system, including for educational purposes..

**Keywords:** educational information systems, information system development methodology, algorithms, client-server application, Vue.js, Vuetify, Vuex, H2, Spring MVC, Java.

## 1 Introduction

The internet and the World Wide Web today contain many useful and interesting tools for solving various problems. Such sites are notably different from ordinary information resources. One difference is the interaction with the user, which appears in the request and processing of data transmitted by the client to solve a specific problem. The content of the pages may depend on the data presented. Another difference is dynamism, obtaining up-to-date data without the need to reload the page. The above provisions constitute the essence of the web application.

Web application development is a multi-step process. Their number and content depends on the specific architecture. Consider the option of developing a knowledge base on algorithms [1].

## 2 Theoretical background and hypotheses

We will go through the entire architecture from the bottom up (Fig. 1), starting with the database. Its tasks will include the storage of data, as well as their provision upon request from the server. Given the features of the subject area, the classical relational data storage model fully satisfies all the needs of the system, therefore, the structural query language (SQL) will act as a means of communication between the server and the database [2]. In a specific implementation, the H2 DBMS is used, which is characterized by wide functionality with a compact size and low resource requirements.

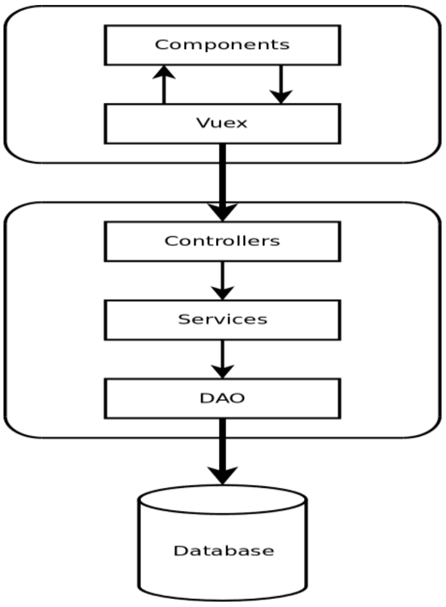
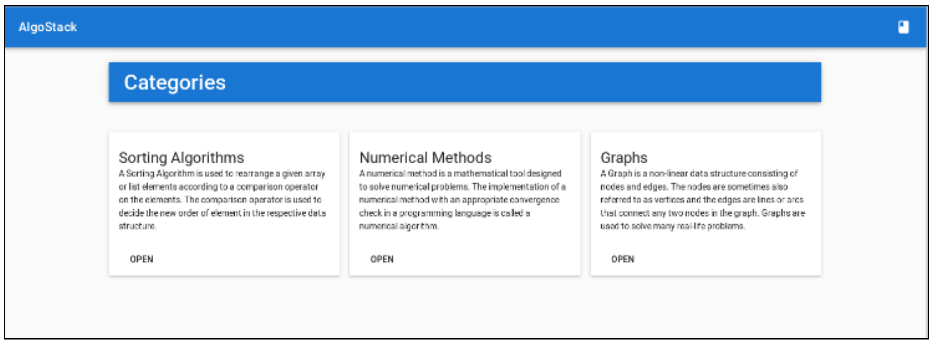


**Fig. 1.** The architecture of the developed application.

Based on the above factors and additions, we will develop a system-dynamic data management model in the Powersim Studio 7.0 application software package that will include the ability to connect and provide additional services and activities that will affect profit (Fig 2) [1].Subsequent paragraphs, however, are indented.



Sorting Algorithms
A Sorting Algorithm is used to rearrange a given array or list elements according to a comparison operator on the elements. The comparison operator is used to decide the new order of element in the respective data structure.

Numerical Methods
A numerical method is a mathematical tool designed to solve numerical problems. The implementation of a numerical method with an appropriate convergence check in a programming language is called a numerical algorithm.

Graphs
A Graph is a non-linear data structure consisting of nodes and edges. The nodes are sometimes also referred to as vertices and the edges are lines or arcs that connect any two nodes in the graph. Graphs are used to solve many real-life problems.

**Fig. 2.** Visitor homepage.

The server component of the application  serves clients, accepting requests from them, responds to them, wherein connects to the database to synchronize the state.

The server is divided into three layers, each of which performs a specific set of functions. We characterize them, indicating the practical application for our work.

The first layer is an interface for working with data. It is built in such a way that a possible switch to another database does not have any effect on other parts of the application. Using this approach, this layer is called a DAO (Data Access Object) or class level.

Work with each of the subject area entities is carried out by creating a special repository that implements the corresponding DAO and has a set of methods for selecting, creating, modifying, or deleting certain data.

The second layer is a set of services. They represent the core business logic of the application. This is where requests from users are processed. Each of the services performs a separate task, while it is granted access only to those repositories that are really needed.

The created application will have the following services: authorization, registration, interaction with the catalog, obtaining information on a separate algorithm, working with sources, administration and others.

The next level is called the controller layer. This is where the primary processing of client's requests is performed. When he accesses the server by calling a specific API method, the request comes to a special dispatcher that has information in the form of a map, in which each existing path corresponds to a method of a controller, and also the availability of this resource for the user is checked.[6]

On the other hand, each method included in the application API has the ability to receive data and send it as a result of the request. In its implementation are calls to individual services, with the transfer of individual information. In the created application there are many controllers, matching in name with services, as a rule.

Thus, the server component plays an important role in the operation of the application. To interact with clients, a special API is provided that describes all the methods, their parameters (which will be returned upon completion of the request processing), as well as its availability.

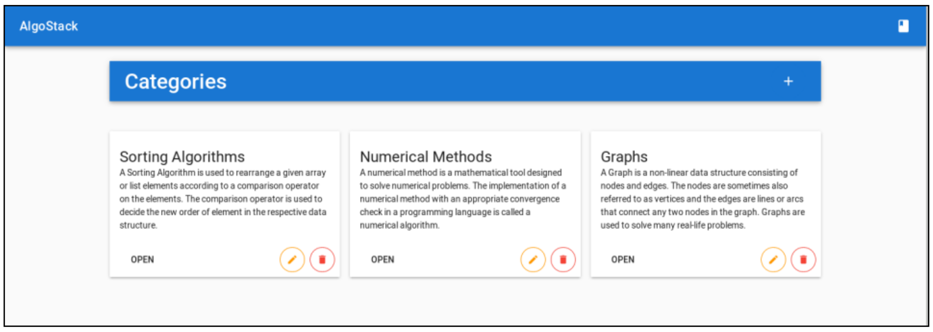
## 3 The method and the empirical results

In structure, the client is divided into individual components. Each of them has its own fields, methods in which events from the user are processed and notifications from other components are received. They can be nested in each other, forming various hierarchies [3].

The structure of each component consists of three parts: a template written in HTML and defining the general structure of the component, that is, how it will be displayed; its state variables, methods, handlers, etc. are described below; the last part, optional, describes the visual component, can be implemented both on pure CSS, and on any preprocessor.

For working with data, or rather their synchronization between components and the server, a special extension is used, the Vuex state manager. In fact, it is a local storage into which data is downloaded from the server, and then distributed to components upon request. For each entity, a special class is created containing four sections. The first section (state) describes what the structure of the stored data will be. The second section (mutations) defines a set of modifications that can be applied to data. It is worth noting that it is impossible to change the data in any other way. The third section (actions) contains descriptions of events that may occur when working with this entity. It is recommended to cause mutations precisely from under the events, which is performed in this work. The fourth section (getters) defines methods for obtaining this or that information on essence has the caching property, that is, the method is executed once, and its result is saved and returned every time the getter is called. If the data on which the result depends has changed, the method is recalculated. Thus, this extension coordinates the work of all components, providing relevant information.[5]

When creating the application, an approach was implemented when everything is in the root component, and the remaining elements that display the different parts of the application are displayed using routing, where a specific address is set for each component. We give examples of various components in the figures below (see Fig.3 and Fig 4).

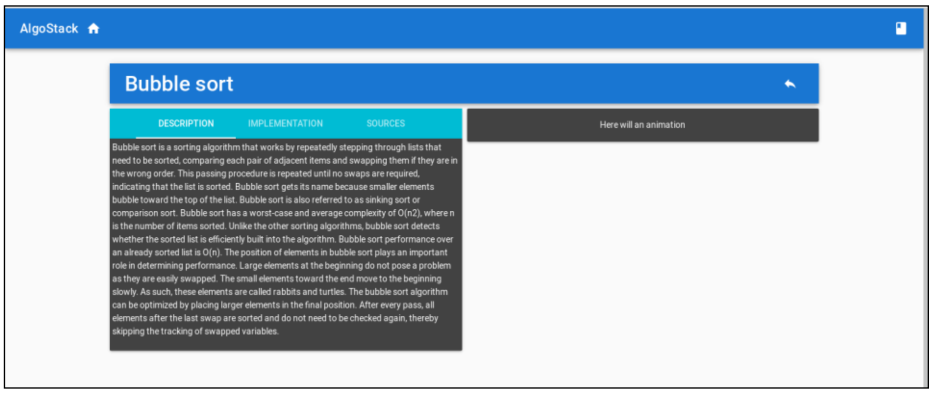


Sorting Algorithms
A Sorting Algorithm is used to rearrange a given array or list elements according to a comparison operator on the elements. The comparison operator is used to decide the new order of element in the respective data structure.

Numerical Methods
A numerical method is a mathematical tool designed to solve numerical problems. The implementation of a numerical method with an appropriate convergence check in a programming language is called a numerical algorithm.

Graphs
A Graph is a non-linear data structure consisting of nodes and edges. The nodes are sometimes also referred to as vertices and the edges are lines or arcs that connect any two nodes in the graph. Graphs are used to solve many real-life problems.

**Fig. 3.** Administrator homepage.



Bubble sort is a sorting algorithm that works by repeatedly stepping through lists that need to be sorted, comparing each pair of adjacent items and swapping them if they are in the wrong order. This passing procedure is repeated until no swaps are required, indicating that the list is sorted. Bubble sort gets its name because smaller elements bubble toward the top of the list. Bubble sort is also referred to as sinking sort or comparison sort. Bubble sort has a worst-case and average complexity of O(n2), where n is the number of items sorted. Unlike the other sorting algorithms, bubble sort detects whether the sorted list is efficiently built into the algorithm. Bubble sort performance over an already sorted list is O(n). The position of elements in bubble sort plays an important role in determining performance. Large elements at the beginning do not pose a problem as they are easily swapped. The small elements toward the end move to the beginning slowly. As such, these elements are called rabbits and turtles. The bubble sort algorithm can be optimized by placing larger elements in the final position. After every pass, all elements after the last swap are sorted and do not need to be checked again, thereby skipping the tracking of swapped variables.

**Fig. 4.** View information on a specific algorithm.

As we see, the distribution of roles is also implemented in the application, according to which a user who has certain privileges has various functionalities available.

Thus, an application that interacts with the database, has functionality divided into two roles: administrator and regular user is derived. The server part is made according to the architecture: DAO - services - controllers, where each layer does its job, starting from connecting and interacting with the database, ending with the processing of user requests. The implementation of the client part is written on the Vue.js framework in the ideology of component construction of the interface, as well as with the presence of the Vuex application state manager, which ensures data synchronization and dynamic user interaction. The application of this project can be integrated in various spheres such as educational, commerce and statistical forecasting purposes [4].[7]

## 4 Conclusion

There are many tasks in the development prospects of this application. Some of them are an animation of the algorithms for greater clarity, the possibility of step-by-step execution of algorithms, as well as custom data sets for testing the application, along with comparing different algorithms according to specified criteria.

One practical implication of this study is the inclusion of this framework in higher educational learning management systems (LMS) for e-learning purposes.

## References

1. Filipova O: Software Development From A to Z. Springer Science, Berlin, Germany (2018).
2. Karpova I.: Databases: A tutorial for the course "Databases". PD MSIEM, Moscow, Russia (2009) (in Russian).
3. Callum Macrae: Vue.js: Up and Running. O’Reilly, Sebastopol, USA (2018).
4. Tarasyev, A. M., Vasilev, J., Turygina, V. F.: Statistical analysis and forecasting of extraction and use of natural resources. AIP Conference Proceedings 2040, 050011 (2018).

5. Lapshina, S. N., Romanovskaia E. M.: The relevance of the use of electronic educational resources in professional education. AIP Conference Proceedings 2116, 430004 (2019).
6. Shangina E.I., Shangin G.A., Ilysheva N.N., Lapshina S.N., Parusheva S.S.: Model of mobile learning and its application in the educational process. AIP Conference Proceedings 1978, 440017 (2018).
   Berg, D. B., Zvereva, O. M., Nazarova, Y. Y., Chepurov, E. G., Kokovin, A. V., Ranyuk, S. V.: Educational network comparative analysis of small groups: Short- and long-term communications. AIP Conference Proceedings 1906, 070012 (2017).