

# Development of a real-time document approval system

*Larisa Cherckesova*<sup>1</sup>, *Nikolay Boldyrikhin*<sup>1</sup>, *Elena Revyakina*<sup>1</sup>, *Olga Safaryan*<sup>1\*</sup>, and *Irina Yengibaryan*<sup>2</sup>

<sup>1</sup>Don State Technical University, Gagarin Square, 1, Rostov-on-Don, 344003, Russia

<sup>2</sup>Noth-Caucasian Branch of Moscow Technical University of Communication and Informatics, 62 Serafimovich Street, Rostov-on-Don, 344002, Russia

**Abstract.** The article describes the preparation of the document approval system before the start of development, the project architecture, and testing. The work processes are demonstrated, registration and authorization in the system, creation of the document approval process and the approval process itself are considered.

## 1 Introduction

In the modern world, the topic of electronic approval of documents is very relevant, as it significantly speeds up the decision-making process on projects, especially if the participants in the approval are geographically located in different places. Information about the process status, documents, and comments is promptly delivered to the participants.

In addition, the introduction of document approval systems into production has a positive impact on both the economic and environmental aspects. Printing and photocopying costs are significantly reduced, which minimizes the use of resources required for printing [1-3].

## 2 Description of the document approval system operation process

To start working in the system, the user must pass authorization or registration, if he is not already authorized in it. After logging in, the user can interact with the approval processes.

For example: upload documents to the system; upload documents from the system; view documents and processes that it has access to in real time; create approval processes; participate in the approval processes of other users; receive notifications about changes in the status of the approval process; upload an automatically generated approval sheetlist.

The document approval scheme starts with the approval stage. If all the clients at the stage have agreed on the document, the stage is assigned the status "Processed successfully", otherwise the status "returned to the author for revision" is assigned. If there

---

\* Corresponding author: [safari\\_2006@mail.ru](mailto:safari_2006@mail.ru)

are more stages, then repeat the procedure described earlier, otherwise we complete the process and assign the status "Document agreed".

The framework was chosen for the development of the web application Vue.js. To speed up the process of creating a project, the Vue-cli library is used. The "vue ui" command, entered in the command-line interpreter (Fig. 1), opens the local web application in the browser. This is how the application is configured [4, 7].

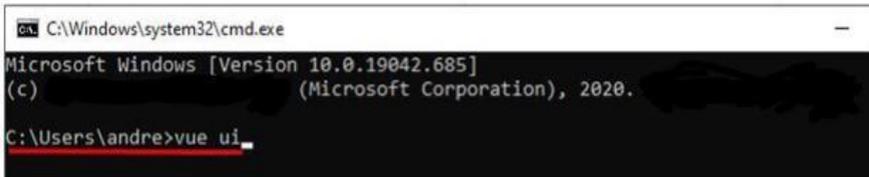


Fig. 1. Command Line Interpreter.

The "create new project" button opens a menu that specifies the project location, name, additional options, and git repository. The Vue cli provides three ways to do this: default (only babel and eslint); manual; remote (download from the git repository).

For the project, the manual method of linking the configuration and plugins was chosen, so the vue cli suggests enabling popular plugins Fig 2. Plugins such as "babel", "router", "vuex", "linter", "vuetify" were selected. Then you need to start creating the project [5].

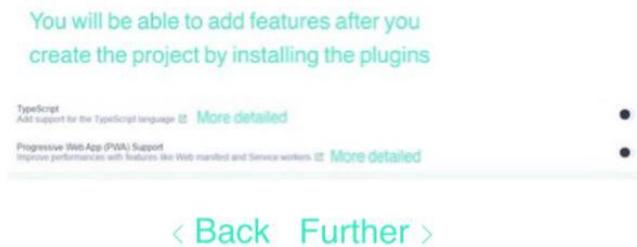
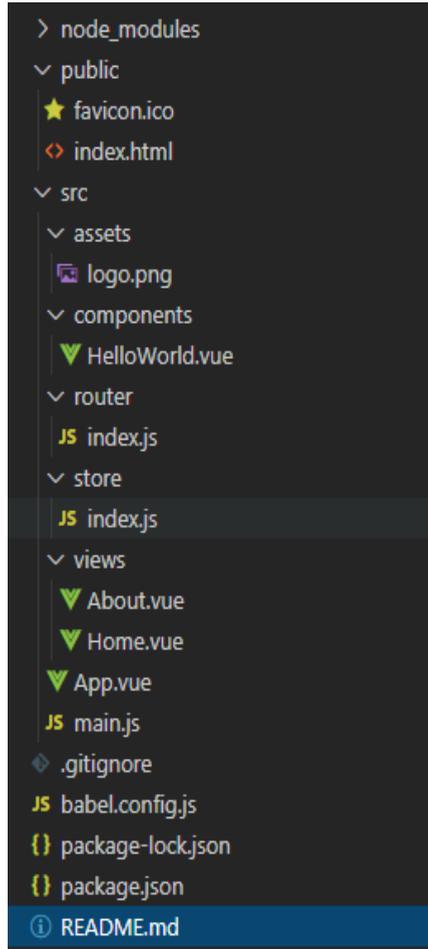
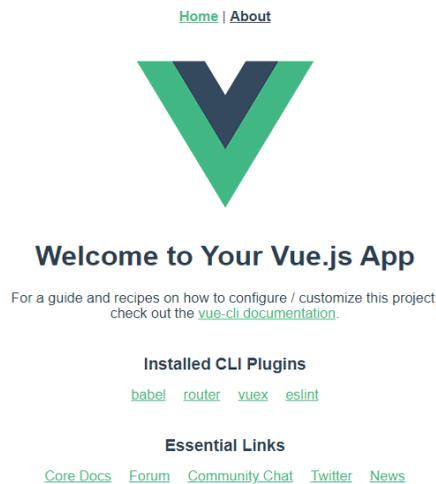


Fig. 2. Selection of plugins.

The final view of the project structure is shown in Fig. 3. The configured initial project is started by entering the "npm run serve" command in the terminal. The project can be opened either locally or over the network (Fig. 4). The initial project view is shown in (Fig.5).



**Fig. 3.** Project structure.



**Fig. 4.** Initial project.

```
- Local: http://localhost:8080/  
- Network: http://192.168.43.23:8080/
```

Fig. 5. Starting a project.

### 3 Application architecture

The application architecture consists of three main parts: client; server; and database. The architecture diagram is shown (Fig. 6). This architecture allows you to implement all the necessary functionality for the goal [6, 8-10].

The client part is implemented on the basis of the Vue framework and its auxiliary libraries. Vue is a modern, open-source JavaScript framework for creating user interfaces that is rapidly evolving. It can easily integrate with many libraries and with existing web applications.

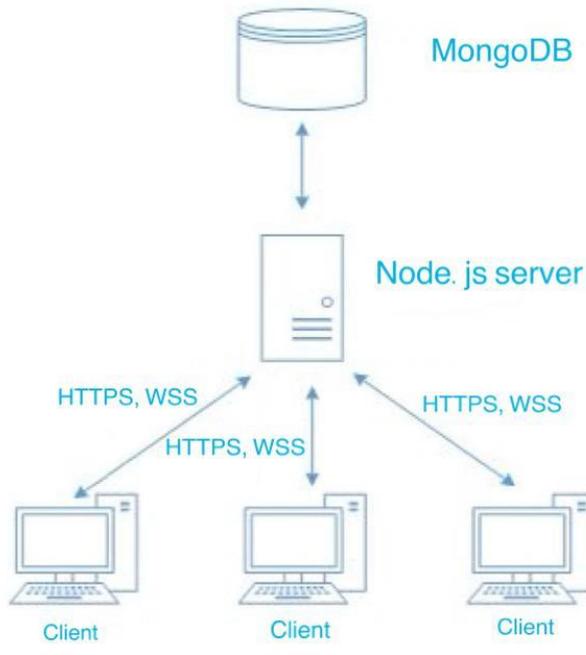


Fig. 6. Application architecture Diagram.

To create a full-fledged website, the framework must be used with modern tools and additional libraries. The most popular of these tools and libraries are: Vuex-a library for managing the state of the application; Vue Router - a library for routing through the site; Vue-devtools-a browser-based developer tool; Vuetify-a user interface library.

When creating large or medium-sized single-page applications, in most cases there is a need to globally monitor the state of the entire application, and not just in a specific Vue component. To solve this problem, the best option is to use the Vuex library, which is a centralized repository for the entire application. Vuex helps you manage your state by writing additional code.

No full-fledged single-page application can do without routing. To implement it on the basis of Vue, you need to use Vue Router library. With Vue-router and Vue, creating routing is quite easy: you need to map components to routes and explain to Vue-router what to display on which path.

Also, one of the tools that a Vue-developer needs is the Vue-devtools extension.n. This extension works in Chrome and Firefox browsers. The Vutify user interface library allows you to significantly save time on development and avoid unnecessary work on ui-ux design and creating many complex components. Vutify is the most popular UI-LIBRARY for Vue. Its goal is to provide everything you need to create rich and interesting web applications using the Material Design specification [7].

The server part is implemented using the Node software platform.js and framework Express.js. Express is a small but flexible web framework for building applications on Node.js, offering a wide range of features for both mobile and web applications. Having at its disposal a considerable number of HTTP service methods and intermediate handlers.

The database consists of the data necessary for the functioning of the document approval system. The MongoDB system was chosen as the data management system. MongoDB is a multi-functional database management system focused on working with documents, using the bson format as data storage.

## **4 Description of the application**

Before using the application for approval of documents, the client must register and log in to the system. If the client is not logged in, the application will redirect him to the registration form. After registration and authorization, the user is taken to a page where it is possible to interact with the document approval processes.

On the page for creating a document approval process, you can: give a name to the approval process; upload the document to be approved; assign people to the various stages of document approval.

In the system, it is possible to create different versions of document approval: parallel approval; sequential approval; mixed approval [8, 12-17].

After loading the document, assigning people to the approval stages, and adding the process name, the "Create Process" button becomes active. When you click on the button, the process information is sent to the server, which processes it and saves it to the database. The client application redirects the user to the created processes page.

All users who are assigned to approve the document are notified that they are participating in the process. On the "Documents to be processed" tab, you can view detailed information about the process that you are participating in. In a process that requires processing, the following information is displayed: the name of the process; the processing status; date created; date of processing. It is possible to download and view a document that requires processing by clicking on the following button.

The "Process" button opens the approval dialog box, where you can make a decision on the document and leave a note.

The document can be approved or returned for revision, after the decision, the "vote" button becomes active. When you click on it, the data is sent to the server for processing. The post-decision process card is shown in. In the user profile of the author of the process, on the "My Processes" tab, you can track the progress of decision-making at the approval stages.

Each stage has one of 4 statuses: pending-indicated in gray; in progress-indicated in yellow and an icon in the form of a clock; agreed-indicated in green; document return-indicated in red. It is also possible to display the status of the approval process in detail. To

do this, click on the "Learn more" button. When you hover over the solutions of other users of the system, detailed information about the solution is displayed.

After successfully completing all the stages of document approval, the author displays a new button on the process card that allows you to download the approval sheet. The approval sheet is shown in Table 1.

**Table 1.** Approval sheet.

Matching	Result	Note	Stage Number	Date and time
Alekseyev Peter	Agreed	-	1	05-12-20 15:25
Fedorov Danil	Agreed	-	2	05-12-20 16:15
Vasin Vasily	Agreed	I accept the document, everything is done according to the requirements	3	05-12-2- 19:05

## 5 Discussion and conclusions

The article describes the scheme of the user's work and the scheme of document approval. The preparation of the document approval system before the start of development, the project architecture, and testing are displayed. The work processes are demonstrated, registration and authorization in the system, creation of the document approval process and the approval process itself are considered.

## References

1. V.V. Repin, Business processes. Modeling, implementation, management (Mann, Ivanov and Ferber, Moscow, 2013)
2. P. Razumov, N. Boldyrikhin, L. Cherckesova, et al., E3S Web of Conferences, **224**, 01033 (2020)
3. V.V. Zhilin, I.I. Drozdova, I.A. Sakharov, et al., in Proceedings of IEEE East-West Design And Test Symposium, EWDTs 2019, Institute of Electrical and Electronics Engineers Inc., 8884375 (2019), DOI: 10.1109/EWDTs.2019.8884375
4. What is Vue, Vuejs, URL: <https://ru.vuejs.org/v2/guide/index.html> (obr date/: 20.04.2020).
5. A.A. Gorbachev, Naukosfera, **4**, 96-99, URL: <https://www.elibrary.ru/item.asp?id=42805918>.
6. L. Bass, P. Clements, R. Katzman, Software architecture in practice, 2nd edn. (Peter, Moscow, 2006)
7. Client-server architecture, Buzza. URL: <https://yandex.ru/turbo/bugza.info/s/architektura-klient-server/> (accessed 18.10.2020).
8. D. Gum, JavaScript for Professionals, 2nd edn. (LLC "I. D. Williams", Moscow, 2016)
9. R. Loose, B. van der Sanden, M. Reniers, R. Schiffelers, IFAC-PapersOnLine, **51(7)**, 381-387 (2018)

10. M. Kulawiak, A. Dawidowicz, M. E. Pacholczyk, *Computers & Geosciences*, **129**, 26-37 (2019)
11. M. AsadSaleem, S.K.H. Islam, S. Ahmed, et al., *Journal of Information Security and Applications*, **58**, 102769 (2020)
12. M.L. Duch, M.R.P. Grossmann, T. Lauer, *Journal of Behavioral and Experimental Finance*, **28**, 100400 (2020)
13. J. Calvo, J. Gracia, E. Bayo, *Advances in Engineering Software*, **112**, 136-146 (2017)
14. S.F. Handayani, E.R. Mahendrawathi, *Procedia Computer Science*, **161**, 1114-1121 (2019)
15. F. Ramadhani, E.R. Mahendrawathi, *Procedia Computer Science*, **161**, 1131-1138 (2019)
16. A. Van Looy, *Information & Management*, **58(2)**, 103413 (2020)
17. R. Salvador, M.V. Barros, L.M. da Luz, et al., *Journal of Cleaner Production*, **250**, 119555 (2020)