

Development of Software Complex for Digitalization of Enterprise Activities

G. T. Balakayeva, K. K. Nurlybayeva, M. B. Zhanuzakov

Abstract—In the proposed work, we have developed software and designed a software architecture for the implementation of enterprise business processes. The proposed software has a multi-level architecture using a domain-specific tool. The developed architecture is a guarantor of the availability, reliability and security of the system and the implementation of business processes, which are the basis for effective enterprise management. Automating business processes, automating the algorithmic stages of an enterprise, developing optimal algorithms for managing activities, controlling and monitoring, reducing risks and improving results help organizations achieve strategic goals quickly and efficiently. The software described in this article can connect to the corporate information system via two methods: a desktop client and a web client. With an appeal to the application server, the desktop client program connects to the information system on the company's work PCs over a local network. Outside the organization, the user can interact with the information system via a web browser, which acts as a web client and connects to a web server. The developed software consists of several integrated modules that share resources and interact with each other through an API. The following technology stack was used during development: Node js, React js, MongoDB, Ngnix, Cloud Technologies, Python.

Keywords—Algorithms, document processing, automation, integrated modules, software architecture, software design, information system.

I. INTRODUCTION

IN recent years, digitalization has been transforming the way enterprises operate, particularly in the areas of e-document approval processes and personnel management. Digitalization refers to the use of digital technologies to automate, streamline, and optimize business processes, enabling organizations to operate more efficiently and effectively. Digitalization contributes to the monitoring and improvement of the management awareness of the chronology of task execution, improvement of the companies' internal and external services by reducing execution time.

One of the key benefits of digitalization is the reduction of paperwork and manual processes, which can be time-consuming and error-prone. E-document approval processes, for example, can be digitized using electronic signature solutions, allowing for faster and more secure approval of important documents. This can significantly improve business operations, particularly in industries that rely heavily on paperwork, such as finance, healthcare, and legal services.

The importance of digitalization of enterprise activities

cannot be overstated. In an increasingly digital world, enterprises that fail to adopt digital technologies risk falling behind their competitors and losing market share. Furthermore, with the COVID-19 pandemic highlighting the importance of remote work, digitalization has become more critical than ever for enterprises to adapt and thrive in a rapidly changing business landscape [1].

II. ARCHITECTURE

In the realm of workflow systems, those centered around document-driven processes exhibit greater flexibility compared to conventional control flow methodologies. They are not only simpler to verify but also prove to be more effective for ad hoc workflows as provided in [2].

The software presented in this study is called "E-enterprise". It is composed of multiple modules, including electronic document management, personnel management, and other subsystems. The subsystems are designed to interact with the main system and micro services through API calls. The document approval service provides digital signatures and generates documents by manipulating data from the "Documents" data collection of the database. Similarly, the rating service analyzes employee data and provides promotion predictions using data from the "Rating" data collection of the database. A kind of Employee Scoring implementation helps to automate the employees promote decision and makes this process transparent and eliminates the human factor. The employee onboarding and dismissal service automates the hiring and dismissal processes and manages data in the "Employee history" collection.

The architecture of the software complex is depicted in Fig. 1. It illustrates the communication between various modules and services within the system.

III. AUTOMATING ENTERPRISE PROCESSES

The personnel management is an essential component of any organization, and its function includes various tasks related to personnel management [3]. In this article, we describe the main functions of the personnel management and the algorithms used for hiring and dismissing employees. We also detail the necessary documents required during the hiring process and the contents of the employee's personal file.

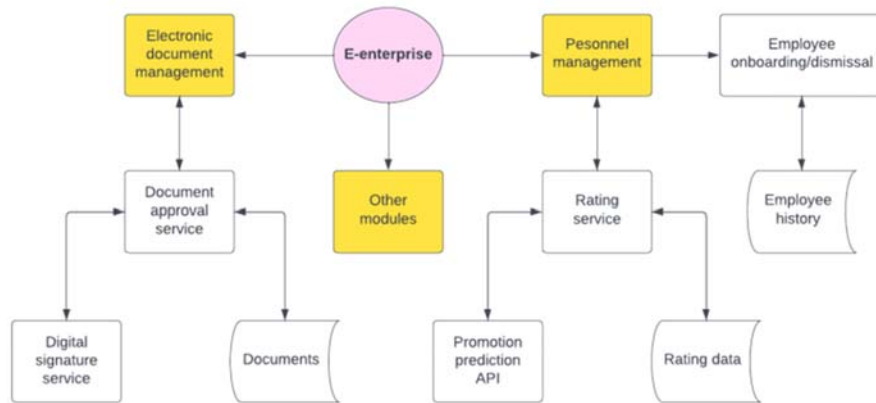


Fig. 1 Architecture of the software complex

Open Science Index, Electrical and Computer Engineering Vol:17, No:12, 2023 publications.waset.org/10013401.pdf

The HR department's primary functions are hiring and dismissal [4]. The algorithm for hiring an employee includes five steps: user registration, hiring a user, drawing up an employment contract, creating and filling out a personal file, and changing the staffing table. On the other hand, the algorithm for dismissing an employee consists of five steps as well: drawing up a dismissal order, changing information about the employee's work activity, issuance of documents, changing the staffing table, and generating a list of dismissed employees.

During the registration process, the employee is required to provide their identity card (passport), work record, and education document. The employee's status is updated to "pending employment" ("waiting_interview_decision") until the management conducts an interview. If the interview is successful, the employee's status is updated to "hired" (in the process of drafting an employment contract). At this stage, an employment contract is drawn up, which includes the employee's surname, first name, patronymic, service number, position (workplace), and basis (number and date of the employment order). After this, a mark is entered into the employment record book, and an employee's personal file is formed.

The employee's personal file contains essential information, including their passport data, general information, personal data, education, labor history, and documents. The personal data include the employee's date of birth, place of birth, address, identity card (passport) number, pension certificate, medical certificate, INN (Individual identification number), marital status, and children. The education section includes information about the employee's education, and the labor history section includes details of the employee's work history, such as organization, position, experience, and article of dismissal. The documents section provides an inventory of documents attached to the employee's personal file.

After the personal contract is completed, the employee's status is updated to "works in the state," and there is a change in the staffing table. In case of an employee's dismissal, a dismissal order is drawn up, and a mark is put in the employee's personal file. The personal file is then transferred to the archive for further storage. After the mark is made in the employee's personal file, information about the employee's work activity is

changed, documents are issued, and the staffing table is updated with a list of dismissed employees.

Overall, the HR department's functions are crucial for the efficient management of an organization's personnel. It is vital not only for the system to incorporate user friendly features but also to cultivate a necessary collaborative culture in the human factor [5]. The algorithms used for hiring and dismissing employees ensure a structured and organized approach. Proper maintenance of the employee's personal file, including the necessary documents and information, is essential for effective personnel management.

The electronic document management module provides two document submission workflows for users: parallel and sequential.

A. Sequential Workflow of the Document

In the sequential workflow, the document is sequentially passed from one executor (a user who reviews and signs the document) to another.

The document object is a representation of the data stored in the database. The executors (users who approve and sign the document) are chosen by the initiator (a user who submits the document) and stored in the "executors" array in the document object. The object also contains a "signed" array, which stores the IDs of the executors who have signed the document. The "currentViewer" property is used to identify the current executor who has permission to view the document, while the "status" property displays the current status of the document.

To implement the sequential workflow of the document, a stack data structure is used. When the document is created, the "executorsStack" property is added to the object, which is created from the reversed "executors" array. The array is popped every time the document is passed to the next executor, and the popped executor is assigned as the "currentViewer". The process ends successfully when the stack is empty, meaning all executors have reviewed the document, and it is complete.

In the best-case scenario (when all executors agree and sign the document), the initiator adds "user1", "user2", and "user3" to the "executors" array, so the initial state of the object is as follows:

```

{
  executors: ["user1", "user2", "user3"],
  executorsStack: ["user3", "user2", "user1"],
  signed: [],
  currentViewer: null,
  status: "processing",
  ...otherProps
}
    
```

When the document is created, the "currentViewer" is assigned the `executorsStack.pop()` value, which, in this case, is "user1". The current executor signs the document with the "sign" method, and the "currentViewer" is then pushed to the "signed" array. In this method, the "executorsStack" is popped, and it is assigned to the "currentViewer" property (in this example, "user2").

The second executor then receives the document and can view it, after which they sign it, and the document is passed to the next executor. The process continues until all executors have signed the document. Once the final executor signs the document, the "register" method is called. If the "executorsStack" is empty, the "register" method sets the "status" to "registered," and the document is stored in the archive list.

If any executor rejects the document, the "reject" method is called, which changes the status of the document to "rejected," clears the "executorsStack" and "signed" arrays, sets "currentViewer" to null, and returns the document to the initiator for corrections. After the initiator corrects the document, the process starts over from the first executor.

The process is terminated either when the deadline is reached or the initiator stops the process manually. The flowchart of the sequential document workflow is presented in Fig. 2.

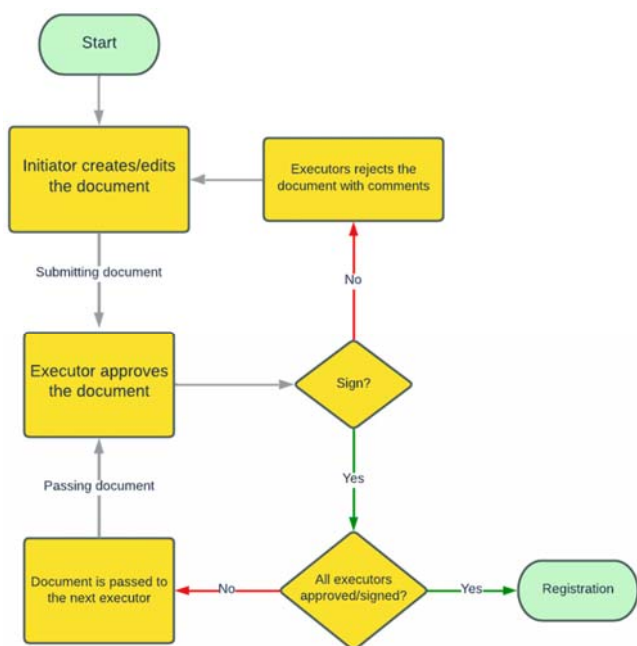


Fig. 2 Flowchart of sequential document workflow

In case an external executor needs to be invited to the

process, the initiator can add their ID to the "executors" array, and the document will be passed to them when it is their turn. The document object's initial state will be the same as in the standard scenario, but with an additional ID in the "executors" array.

B. Parallel Workflow of the Document

In parallel workflow, all the executors have permissions to view/sign the document at any time. The order of approvals does not matter. The only condition to consider is the deadline.

There are many other internal business processes could be implemented in this system, such as Employee Transfers, Business trips, Holidays, sick leave processing, User access management, etc. All of them could gather information from other systems of the enterprise and show the availability of the employee and appoint a substitute in case of necessity. As the system "E-enterprise" is scalable, any new business-processes could be automated and integrated in it.

IV. TECHNOLOGIES

The current study presents a full-stack web application designed to deliver a seamless digital experience for enterprises. The application integrates various technologies that offer unique advantages and disadvantages.

The backend server application is built using Node.js, an open-source, cross-platform runtime environment that leverages Chrome's V8 JavaScript engine. Node.js is fast and efficient, which makes it a suitable choice for developing high-performance web applications [6].

For the client application, the study uses React.js, a JavaScript library designed by Facebook for building user interfaces. React.js provides developers with the capability to build complex UI components while optimizing web page rendering using a virtual DOM, thus improving application performance [7].

The study employs MongoDB as the database management system to store documents and employee information. MongoDB is a NoSQL database that offers scalability, flexibility, and dynamic schemas. It can handle large amounts of data with ease and eliminates the need for complex migrations. In [8], it is concluded that one can choose MongoDB instead of MySQL if the application is data intensive and stores many data and queries lots of data. As the developed software works with Big Data, this technology has been chosen to provide better performance when reading employee documents.

Furthermore, Nginx was used as the web-server handler due to its high-performance capabilities that allow it to handle large traffic volumes while still being highly configurable and customizable to meet specific application requirements. Additionally, it has a small memory footprint, making it an ideal choice for resource-constrained environments [9].

We also used cloud technologies to provide on-demand access to computing resources and services. Cloud technologies offer scalability, cost-effectiveness, and flexibility, as they eliminate the need for expensive hardware and infrastructure.

Finally, we employed Python to develop a microservice for

predicting employee promotion using a pre-built machine learning prediction model. Python is a high-level programming language known for its ease of use and simplicity [10].

Overall, the technologies used in this study were chosen based on their performance, scalability, flexibility, and ease of use. The combination of Node.js and React.js provides a robust front-end framework, MongoDB offers a flexible and scalable database solution, Nginx provides a high-performance web server, and cloud technologies provide on-demand access to computing resources and services.

V.CONCLUSION

In this work, we have presented a software architecture and software design for the implementation of enterprise business processes. The proposed application complex guarantees availability, reliability, and security of the system and the implementation of business processes, which are the basis for effective enterprise management. The article discusses the main functions of personnel management and the algorithms used for hiring and dismissing employees. The proposed software can help organizations achieve strategic goals quickly and efficiently, reducing risks and improving results. The software architecture presented in this article can provide a basis for further research and development in enterprise business process management.

ACKNOWLEDGMENT

This research was funded by the Science Committee of the Ministry of Education and Science of the Republic of Kazakhstan (Grant #AP09259208)

REFERENCES

- [1] Kmecová I., Stuchlý J., Sagapova, N., Tlustý M.. SMA human resources management digitization: evaluation of the level of digitization and estimation of future developments. 2021. Polish journal of management studies. 233 pp.
- [2] Wang, J., Kumar, A. 2005. A Framework for Document-Driven Workflow Systems. In: van der Aalst, W.M.P., Benatallah, B., Casati, F., Curbera, F. (eds) Business Process Management. BPM 2005. Lecture Notes in Computer Science, vol 3649. Springer, Berlin, Heidelberg. 285–301pp. https://doi.org/10.1007/11538394_19
- [3] Cynthia A. Lengnick-Hall, Tammy E. Beck, Mark L. Lengnick-Hall. 2011. Developing a capacity for organizational resilience through strategic human resource management, Human Resource Management Review. 243-255 pp. <https://doi.org/10.1016/j.hrmr.2010.07.001>.
- [4] Dianna L. Stone, Diana L. Deadrick. 2015. Challenges and opportunities affecting the future of human resource management, Human Resource Management Review. 139-145pp. <https://doi.org/10.1016/j.hrmr.2015.01.003>.
- [5] Vlasios Sarantinos. 2008. Considering Document Workflow Issues: Pros and Cons for the Non-experts. International Journal of Business and Management, Vol. 3, No. 7. 3-5pp.
- [6] S. Tilkov and S. Vinoski. 2010. Node.js: Using JavaScript to Build High-Performance Network Programs in IEEE Internet Computing, vol. 14, no. 6, pp. 80-83, doi: 10.1109/MIC.2010.145.
- [7] Sanchit Aggarwal. 2018. Modern Web-Development using ReactJS. Sanchit Aggarwal et al. International Journal of Recent Research Aspects ISSN: 2349-7688. 133-137pp.
- [8] C. Györödi, R. Györödi, G. Pecherle and A. Olah. 2015. "A comparative study: MongoDB vs. MySQL. 13th International Conference on Engineering of Modern Electric Systems (EMES), Oradea, Romania, 1-6pp. doi: 10.1109/EMES.2015.7158433.
- [9] Reese, W. 2008. Nginx: the high-performance web server and reverse proxy. Linux Journal, 2p.

- [10] Srinath, K. R. 2017. Python—the fastest growing programming language. International Research Journal of Engineering and Technology, 4(12), 354-357pp.