E-ISSN: 2962-6110
P-ISSN: 2964-7843
JSRET
jsret.knpub.com
Journal of Scientific Research, Education, and Technology

Development of a REST API for Human Resource Information System for Employee Referral Management Domain Using the Express JS Framework and Node.js

Varinia Azkarin, Rangga Gelar Guntara, Oding Herdiana
Program Studi Bisnis Digital, Universitas Pendidikan Indonesia
Correspondent Email: variniaazkarin@upi.edu

ABSTRACT

This paper presents the development process of a REST API for a Human Resource Information System (HRIS) focusing on employee referral management. The API was developed using the Express JS framework and Node.js, leveraging their powerful capabilities for efficient web application development. The HRIS serves as a centralized platform for managing employee referrals, tracking candidate progress, and optimizing talent acquisition. The article provides insights into the methodology, key features, and advantages of using the Express JS framework and Node.js for building such an HRIS. The results of the study indicate that the Express JS and Node.js frameworks have successfully developed a REST API for HRIS in the domain of employee referral management. This is evident from the black box testing results, which show valid outcomes for each tested request and response in every API endpoint.

Keywords: *REST API, Human Resource Information System, Express JS, Employee Referral Management Domain, Node.js*

INTRODUCTION

In today's highly competitive job market, organizations are constantly seeking innovative ways to attract top talent efficiently and effectively (Masenya, 2022). One proven strategy that has gained significant popularity is the utilization of employee referrals. Employee referrals not only help organizations identify potential candidates with the right skills and cultural fit but also contribute to higher employee engagement and retention rates (Chawla, 2019).

To streamline and optimize the employee referral process, a robust Human Resource Information System (HRIS) is essential. The HRIS serves as a centralized platform for managing various human resource tasks, including employee referral management (Abuhantash, 2023). It enables HR personnel to track referrals, evaluate candidate progress, and make data-driven decisions, ultimately enhancing talent acquisition and internal growth (Gusain et al., 2023).

In this paper, researcher explore the development of a REST API-based HRIS specifically designed for Employee Referral Management, utilizing the powerful combination of the Express JS framework and Node.js. Express JS is a lightweight and flexible web application framework for Node.js (Aung et al., 2022), offering a range of features and tools for building scalable and efficient RESTful APIs. Node.js, on the other hand, provides a runtime environment for server-side JavaScript execution, enabling fast and event-driven development (Challapalli et al., 2021).

By leveraging the capabilities of Express JS and Node.js, organizations can create an HRIS that not only simplifies the employee referral process but also integrates seamlessly with existing HR systems and provides valuable insights through analytics. This paper delves into the methodology, key components, and advantages of developing an RESTFUL API for HRIS for Employee Referral Management Domain using the Express JS framework and Node.js.

In a study conducted by (Suharti & Sulistyo, 2018), the researchers focused on various aspects such as system adoption, user satisfaction, and system performance. They explored the benefits of HRIS implementation in terms of increased efficiency, improved data accuracy, and streamlined HR processes. However, while the study provided valuable insights into the general implementation and impact of HRIS, it did not specifically address the unique domain of Employee Referral Management within HRIS. Employee referrals play a crucial role in talent acquisition, and organizations are increasingly recognizing their significance in identifying high-quality candidates (Schlachter & Pieper, 2019). The lack of research specifically investigating the development and implementation of an HRIS with a focus on Employee Referral Management is a noticeable research gap. There is a need to delve deeper into the specific challenges and opportunities associated with building an HRIS that optimizes the employee referral process, tracks candidate progress, and enhances talent acquisition strategies.

By addressing this research gap, scholars and practitioners can gain a better understanding of how the Express JS framework and Node.js can be effectively utilized to develop a REST API for HRIS specifically tailored for Employee Referral Management. This research would provide valuable insights into the unique features and functionalities required for such a system, as well as explore the benefits it offers in terms of improving the quality of hires, increasing employee engagement, and promoting internal growth.

METHOD

The research process consisted of problem identification, literature review, API design and implementation, black box testing, and drawing research conclusions.

1. Problem Identification

The research process began with problem identification, which was the absence of a backend application and database management system that facilitates data management processes. Therefore, a REST API is required as an intermediary between the frontend and server in the data exchange process.

2. Literature Review

This phase was carried out to examine relevant theories related to the research. Additionally, information or data collection was also conducted through observations of organizations.

3. API Design and Implementation

This phase discusses the design of the REST API that will be built in the meeting room reservation system of organization. This step involves designing each endpoint, including the URI and parameters used. Subsequently, the API calls are implemented using the Postman application (Ahmad et al., 2021). The design and implementation process utilize the HTTP protocol, with the GET method for retrieving data, the POST method for sending data, the PUT method for modifying data, and the DELETE method for deleting data from the database.

4. Testing

After designing and implementing, the testing phase is carried out using the black box method. Black box testing focuses on the functionality of the application. This phase is conducted to assess the compatibility between the input provided and the expected output.

5. SDLC Method

The software development method is also referred to as the Software Development Life Cycle (SDLC). In this study, the author utilizes the waterfall method of SDLC. According to (Guntara, 2022), the waterfall method generally consists of several stages, including: (1) planning and visualization; (2) requirements analysis; (3) software modeling and design; (4) coding; (5) documentation; (6) testing; and (7) deployment and maintenance. An overview of the waterfall method is depicted in Figure 1.

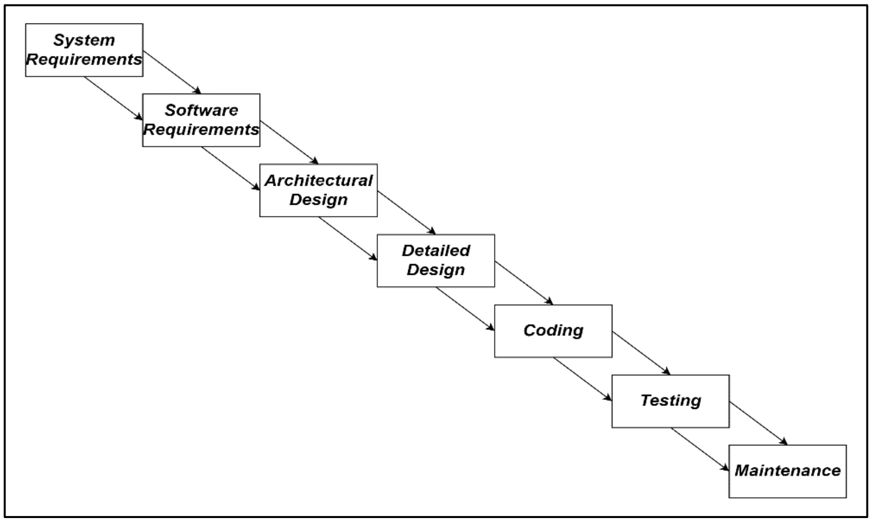


Figure 1: Waterfall Model of SDLC

FINDING AND DISCUSSION

1. Architecture System

This stage aims to provide a general overview of the system's operation, as shown in Figure 2. There are three main components in this architecture: the client-side HRIS application accessed by users through a browser, the Web Service API, and the Database as the data storage. In each request process, users will send requests from the client-side application using the HTTP protocol with methods such as GET, POST, PUT, or DELETE to the Web Service API. Through the implemented program, these requests are processed within the API and forwarded to the Database. The process can involve retrieving data from the Database, inputting data, or modifying data. The API responds with JSON messages indicating whether the data has been successfully sent or provides error messages if there are any issues during the request process.

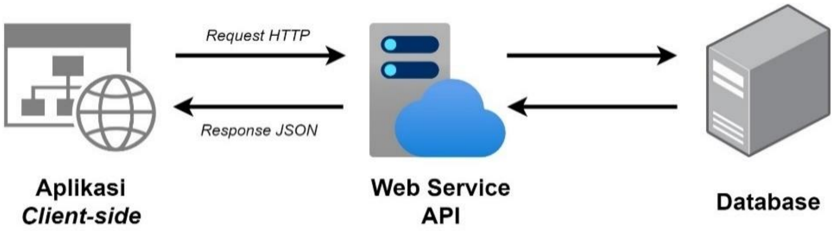


Figure 2: Three Main Components of Architecture System

2. Analysis of Application Users

To support the design of this API, an analysis of the expected user characteristics is required. From the end-user perspective of this HRIS API, which is IT developers, the desired characteristics are users who can understand the usage of the API through the HTTP protocol with the specified JSON input format and comprehend the processing of JSON outputs from the API.

3. Functional Requirements Analysis

Functional requirements analysis is a crucial step in software development to identify and document the specific functionalities that a system or application must possess. This analysis helps ensure that the software meets the needs and expectations of its users.

In this phase, analysis will be conducted through system modeling and visualization. The modeling approach utilized will be object-oriented modeling. To carry out this modeling, the author employs the Unified Modeling Language (UML).

a. Use Case Diagram

In the following Figure 3 is the use case diagram of the HRIS API for employee referral management domain that has been developed.

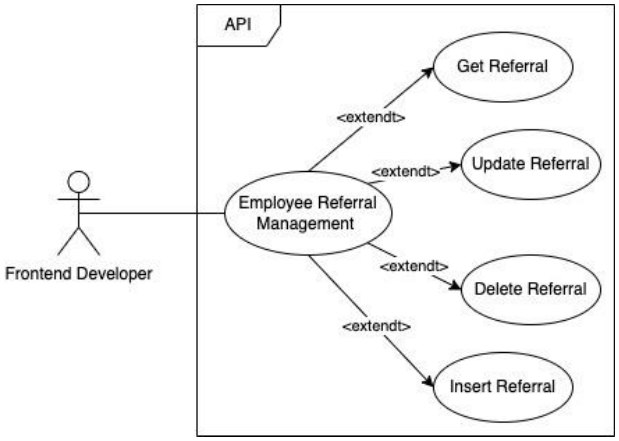


Figure 3: Use Case Diagram Employee Referral Management Domain

b. Use Case Scenario

In the following Figure 4, the sequential diagram of the HRIS API for user management domain that has been developed are presented.

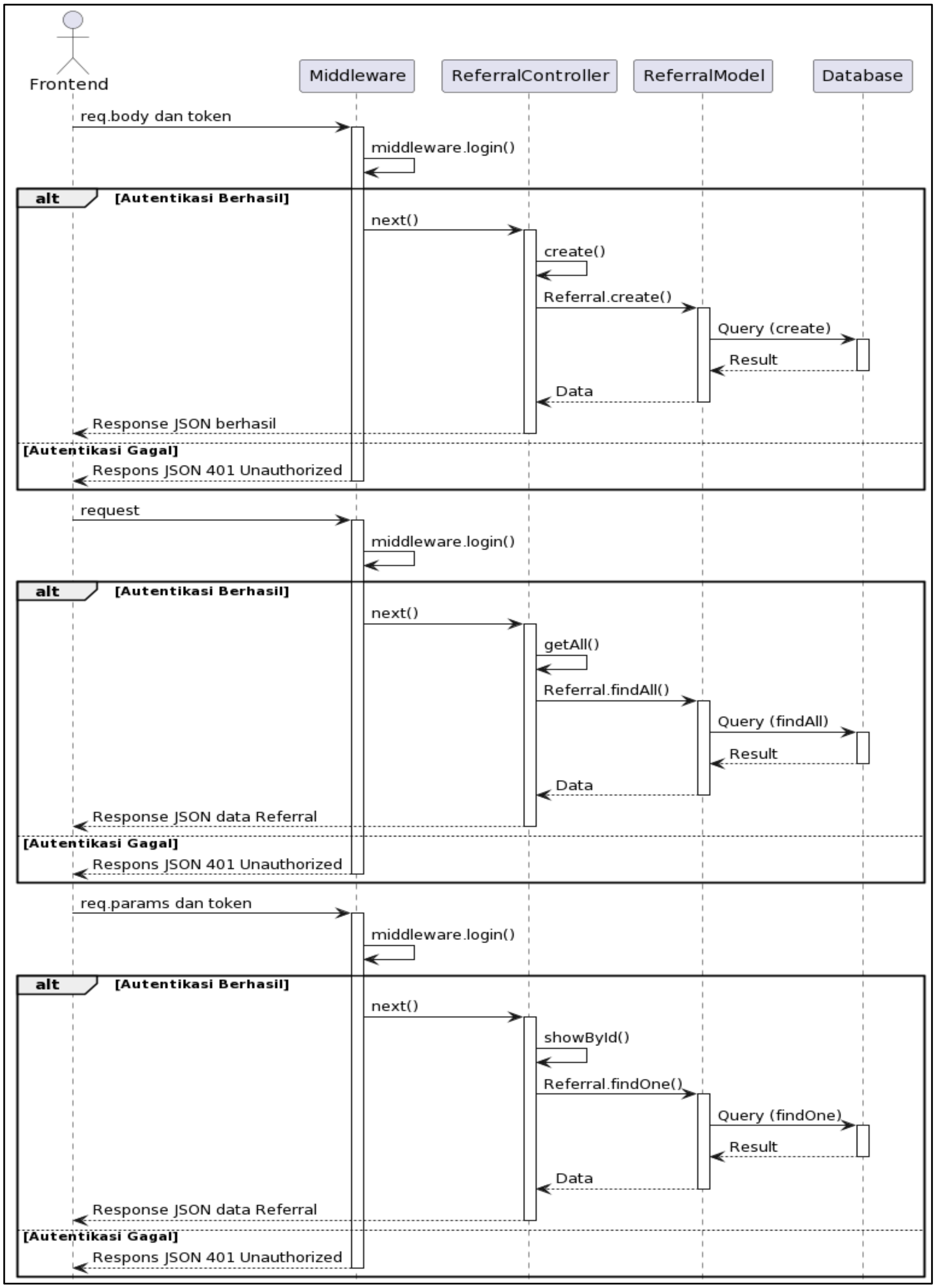


Figure 4: Sequential Diagram Employee Referral Management Domain

4. Database Design

Database design is one of the crucial stages in the design process as it aims to determine how each table is created and how each field can relate to other fields. In the following Table 1, the database design for uemployee referral management domain that has been developed are presented.

Table 1: Database Design

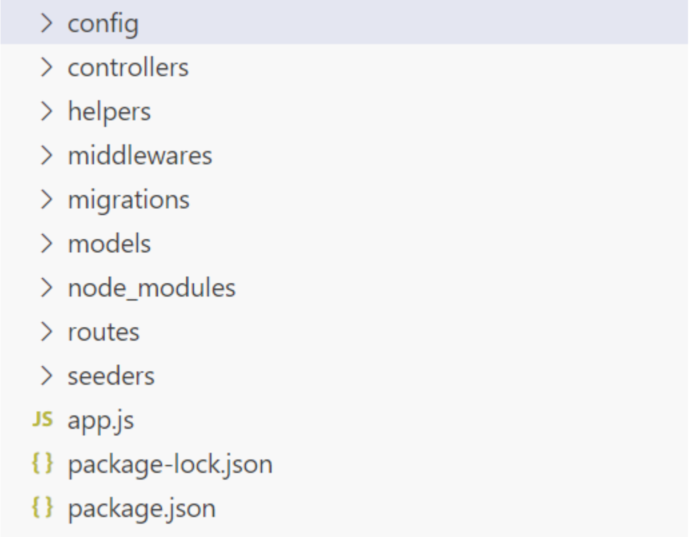
No	Field (Kolom)	Tipe Data	*Key*
1	id	integer	*Primary Key*
2	user_id	integer	*Foreign Key*
3	recruit_id	integer	*Foreign Key*
4	candidate_name	varchar(255)	-
5	details	varchar(255)	-
6	status	boolean	-
7	createdAt	timestamp	-
8	updatedAt	timestamp	-

5. Designing the JSON Response Structure

This phase involves designing the structure of the JavaScript Object Notation (JSON) that will be used. When making an API request, a frontend developer will receive data that has been processed in JSON format. This JSON data is used by the frontend developer to display information to the user on the integrated web page using specific API endpoints. In the JSON response sent by the API, there are three main keys: status, message, and data. The status key contains a value of true or false, indicating whether the request was successful or not. If the request is successful, the status value will be true. The message key contains a response message. The data key contains the requested data from the user, and it is structured as an array of objects.

6. Express JS Implementation

To utilize this framework, several directories are required. Each directory has its own role in the flow of the API program. The directory structure of the HRIS API in this study can be seen in the following Figure 5.



```
> config
> controllers
> helpers
> middlewares
> migrations
> models
> node_modules
> routes
> seeders
JS app.js
{ } package-lock.json
{ } package.json
```

Figure 5: Directory Structure of Express JS

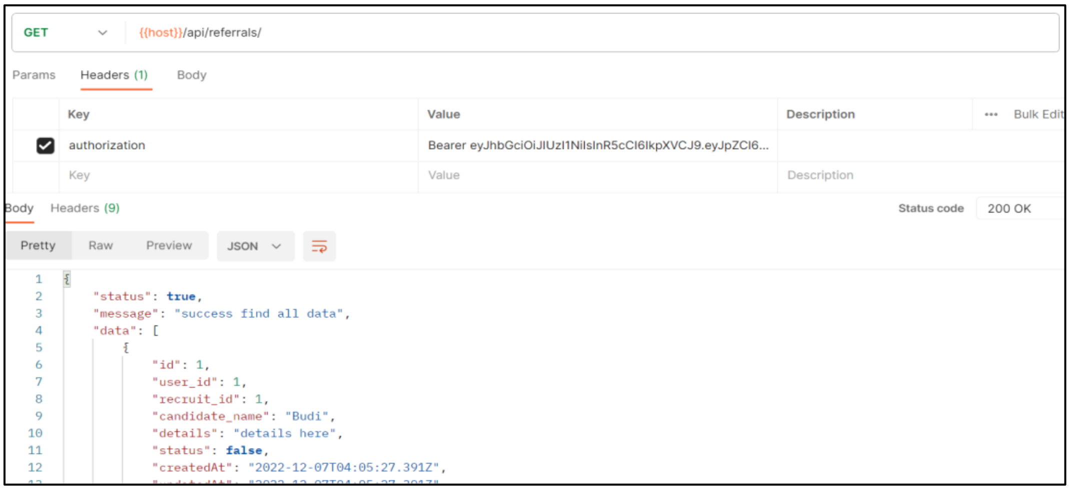
7. API Implementation

The results of API implementation are demonstrated through a detailed table of endpoints and screenshots showcasing the HTTP response results using the Postman application. The details of the API implementation results are as follows:

Table 2: API Endpoint

No.	Metode	URI	Body	Parameter
1	POST	/api/referral	recruit_id, candidate_name, details	-
2	GET	/api/referral	-	-
3	GET	/api/referral/:id		ID *referral*
4	PUT	/api/referral/:id	recruit_id, candidate_name, details	ID *referral*
5	PUT	/api/referral/:id	-	ID *referral*
6	DELETE	/api/referral/	id (array)	-

Here are examples of calling API endpoints using the Postman application, including the request and response for each:



GET {{host}}/api/referrals/

Params | Headers (1) | Body

	Key	Value	Description
☑	authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6...	
	Key	Value	Description

Body | Headers (9) — Status code 200 OK

Pretty | Raw | Preview | JSON

```
{
    "status": true,
    "message": "success find all data",
    "data": [
        {
            "id": 1,
            "user_id": 1,
            "recruit_id": 1,
            "candidate_name": "Budi",
            "details": "details here",
            "status": false,
            "createdAt": "2022-12-07T04:05:27.391Z",
```

Figure 6: API Request and Response

8. Alpha Testing

In the software development life cycle, alpha testing is conducted within the internal environment. For this testing, black box testing is performed, which focuses solely on the application's functionality. The purpose of this testing is to ensure that the application's input and output align with the expected scenarios. The results of the black box testing for the REST API HRIS are presented in Table 3.

Table 3: Blacbox Testing Result

No.	Testing Scenarios	*Test Case*	Result
1	Insert referral	Input key JSON recruit_id, candidate_name, dan details	Valid
2	Get referral	Input URI /api/referral	Valid
3	Get single data referral	Input URI /api/referral/:id	Valid
4	Update referral	Input value key JSON recruit_id, candidate_name, atau details	Valid
5	Approve referral	Input URI /api/referral/accept-referral/:id	Valid
6	Delete referral	Input ID referral array	Valid

CONCLUSION

In this study, a REST API for Human Resource Information System (HRIS) with a focus on employee referral management has been successfully developed using the Express JS framework and Node.js. Through this research, it has been found that the utilization of the Express JS and Node.js frameworks in the development of the REST API HRIS brings significant benefits. The Express JS framework enables rapid and efficient development, while Node.js allows for handling multiple connections effectively. In the context of HRIS, the developed REST API facilitates integration with other applications and services, enhancing efficiency and interoperability in human resource management.

REFERENCES

Abuhantash, A. (2023). The Impact of Human Resource Information Systems on Organizational Performance: A Systematic Literature Review. *European Journal of Business and Management Research*, *8*(3), 239–245. https://doi.org/10.24018/ejbmr.2023.8.3.1992

Ahmad, I., Suwarni, E., Borman, R. I., Asmawati, Rossi, F., & Jusman, Y. (2021). Implementation of RESTful API Web Services Architecture in Takeaway Application Development. *2021 1st International Conference on Electronic and Electrical Engineering and Intelligent System (ICE3IS)*, 132–137. https://doi.org/10.1109/ICE3IS54102.2021.9649679

Aung, S. T., Funabiki, N., Aung, L. H., Htet, H., Kyaw, H. H. S., & Sugawara, S. (2022). An Implementation of Java Programming Learning Assistant System Platform Using Node.js. *2022 10th International Conference on Information and Education Technology (ICIET)*, 47–52. https://doi.org/10.1109/ICIET55102.2022.9779047

Challapalli, S. S. N., Kaushik, P., Suman, S., Shivahare, B. D., Bibhu, V., & Gupta, A. D. (2021). Web Development and performance comparison of Web Development Technologies in Node.js and Python. *2021 International Conference on Technological Advancements and Innovations (ICTAI)*, 303–307. https://doi.org/10.1109/ICTAI53825.2021.9673464

Chawla, P. (2019). Impact of employer branding on employee engagement in business process outsourcing (BPO) sector in India: mediating effect of person–organization fit. *Industrial and Commercial Training*, *52*(1), 35–49. https://doi.org/10.1108/ICT-06-2019-0063

Guntara, R. G. (2022). Aplikasi Pengenalan Citra Wajah di KTP Menggunakan Google Cloud Vision API dan Kairos API Berbasis Android. *ILKOMNIKA: Journal of Computer Science and Applied Informatics*, *4*(2), 198–207. https://doi.org/10.28926/ilkomnika.v4i2.504

Gusain, A., Singh, T., Pandey, S., Pachourui, V., Singh, R., & Kumar, A. (2023). E-Recruitment using Artificial Intelligence as Preventive Measures. *2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*, 516–522. https://doi.org/10.1109/ICSCDS56580.2023.10105102

Masenya, T. M. (2022). *Integrating Talent and Knowledge Management Practices in the New Normal Business Environment* (pp. 113–144). https://doi.org/10.4018/978-1-7998-8451-4.ch006

Schlachter, S. D., & Pieper, J. R. (2019). Employee referral hiring in organizations: An integrative conceptual review, model, and agenda for future research. *Journal of Applied Psychology*, *104*(11), 1325–1346. https://doi.org/10.1037/apl0000412

Suharti, L., & Sulistyo, P. R. (2018). The implementation of human resources information system and it's benefit for organizations. *Diponegoro International Journal of Business*, *1*(1), 1. https://doi.org/10.14710/dijb.1.1.2018.1-7