DOI 10.70286/EOSS-18.05.2026.005.248-259

INFORMATION SUPPORT AND ARCHITECTURE OF AN INTELLIGENT PLATFORM FOR HUMAN CAPITAL MANAGEMENT IN IT PROJECTS

Ziuziun Vadym
PhD, Associate Professor
ORCID: 0000-0001-6566-8798
Korytnyi Oleksii
Master's Deg. Stud.
ORCID: 0009-0006-3453-3077
Department of Technologies Management
Taras Shevchenko National University of Kyiv, Ukraine

Abstract. The article addresses the transformation of traditional HRM systems into intelligent decision-support platforms capable of operating under conditions of high volatility in the IT labor market. The author proposes the concept of «ethical automation», in which multi-objective optimization algorithms are based not only on the technical compatibility of specialists but also on the preventive modeling of occupational burnout risks through strict adherence to an 85 % workload threshold.

A mechanism for multi-criteria team formation is presented, integrating hard and soft skills analysis with dynamic monitoring of developer's psycho-emotional states. The proposed solution enables the conversion of a human-centered approach to personnel management into measurable economic efficiency by reducing operational costs associated with recruitment and talent retention.

Key words: intelligent HRM system, multi-criteria team optimization, work-life balance algorithms, preventive burnout modeling, IT project management, human capital management in IT.

Introduction. The current state of development in both the global and domestic IT industries is characterized by a paradoxical gap: despite the rapid integration of artificial intelligence into production cycles, management systems for the most critical asset, people, remain largely limited to administrative and accounting functions. This issue is particularly important in the context of project teams, which the author has previously investigated [1]. Traditional HRM approaches have proven largely ineffective in the face of a market operating in a permanent «firefighting» mode, marked by a shortage of qualified professionals and critically high levels of employee emotional exhaustion driven by external destabilizing factors.

The core problem lies in the absence of tools capable of predicting team inefficiency or destructive changes in the psycho-emotional state of specialists before they translate into financial losses for the organization. Existing software solutions

often rely on overly mechanistic algorithmizing, which is perceived by creative teams as a control mechanism rather than a form of support.

Accordingly, there is a pressing need for the development of intelligent systems that ensure a balance between mathematically precise resource allocation and the preservation of a healthy, sustainable working environment.

The purpose of this work is to develop and provide a scientific justification for the architecture of an intelligent HRM system that enables multi-criteria project team formation based on the synthesis of professional competencies, project roles, and personal attributes of specialists. The study is focused on implementing mechanisms for automated workload monitoring that ensure the preservation of developers work-life balance as a strategic factor in maintaining the long-term competitiveness of IT businesses.

Research implementation. We consider the description of algorithms for analyzing and evaluating human resources and their allocation across projects.

All systems operate according to a clearly defined sequence of steps that together form an algorithm and the processes through which users perform their tasks. Even a simple «if–then» rule represents a sequence and a mechanism for responding to specific events. For example, the 85 % rule acts as a constraint that prevents the selection module from assigning employees whose workload is greater than or equal to 85 %.

In essence, algorithms represent business rules and the underlying concept according to which end users operate. However, to design such systems properly, it is necessary to understand input points, events, and output points. In addition to the «best-case» scenario, all alternative paths must be considered, including those arising from errors, deviations, or other conditions caused either by the user or the system. Accounting for these factors is essential for robust system design.

The primary algorithm of the HRM system is the search and formation of a project team based on filters defined by the project manager, who must specify roles, team composition, required technical skills (including domain experience), number of team members, and the estimated project duration.

Fig. 1 illustrates the main algorithm for the process of searching and forming a project team. It consists of interaction between the user and a web platform, which uses an API to communicate with the employee selection module. This module, in turn, executes SQL scripts to generate queries to the database. As a result, the project manager is presented with a visual representation of employees assigned to each required role. If no suitable employee is found, the corresponding role icon is highlighted in red, displaying the message: «Employee not found. Would you like to search for an alternative?». By clicking this message, the system initiates an alternative employee search process.

Another key process is the administration of employee information. For example, a front-end developer may be promoted to a senior level and acquire new qualifications. Within the personal profile on the web platform, the employee is able to add updated information regarding skills and competencies. It is important to note that changes to an employee’s level and role are restricted to HR personnel and system administrators only. As a result of any updates, the employee’s profile on the web platform will reflect

the revised information, and the corresponding records in the database will also be updated accordingly. This process is illustrated in Fig. 2.

The processes described above involve user interaction; however, algorithms primarily represent sequences of steps defining interactions between system components. Let us consider an example of interaction between the database and the employee selection module. It is necessary to account for all side system responses to potential errors. The algorithm of this process is illustrated in Fig. 1.

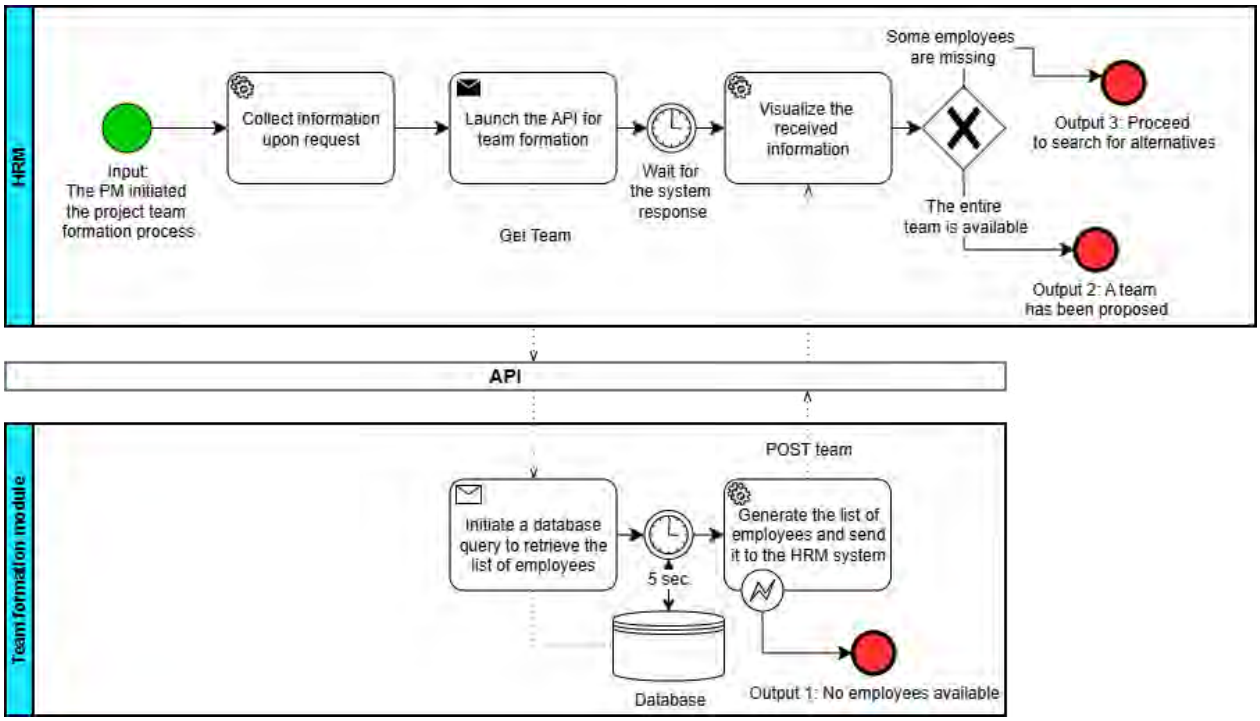

Figure 1. Employee selection algorithm for a project

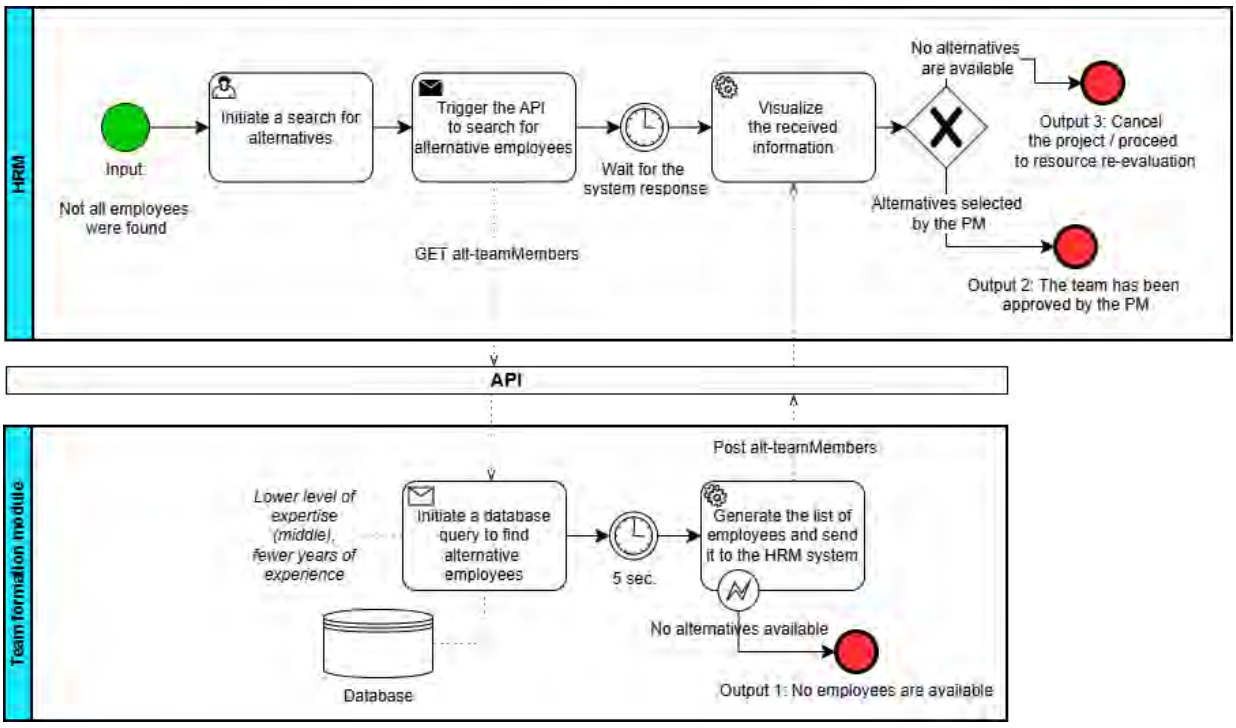

Figure 2. Search for alternative employee’s

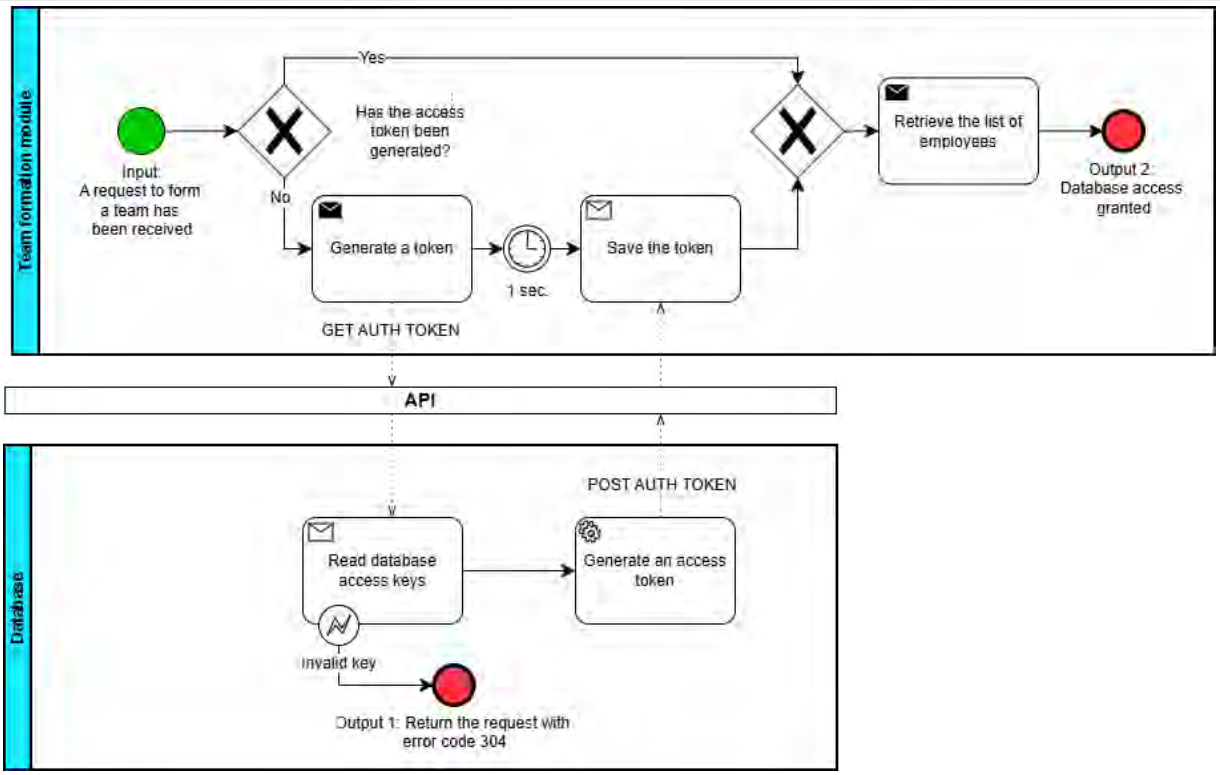

Figure 3. Database access configuration

Since the HRM system is a complex entity encompassing multiple algorithms, the remaining key processes are presented in Fig. 4–5. These include employee information administration (Fig. 4) and the generation of analytical reports on resource utilization (Fig. 5).

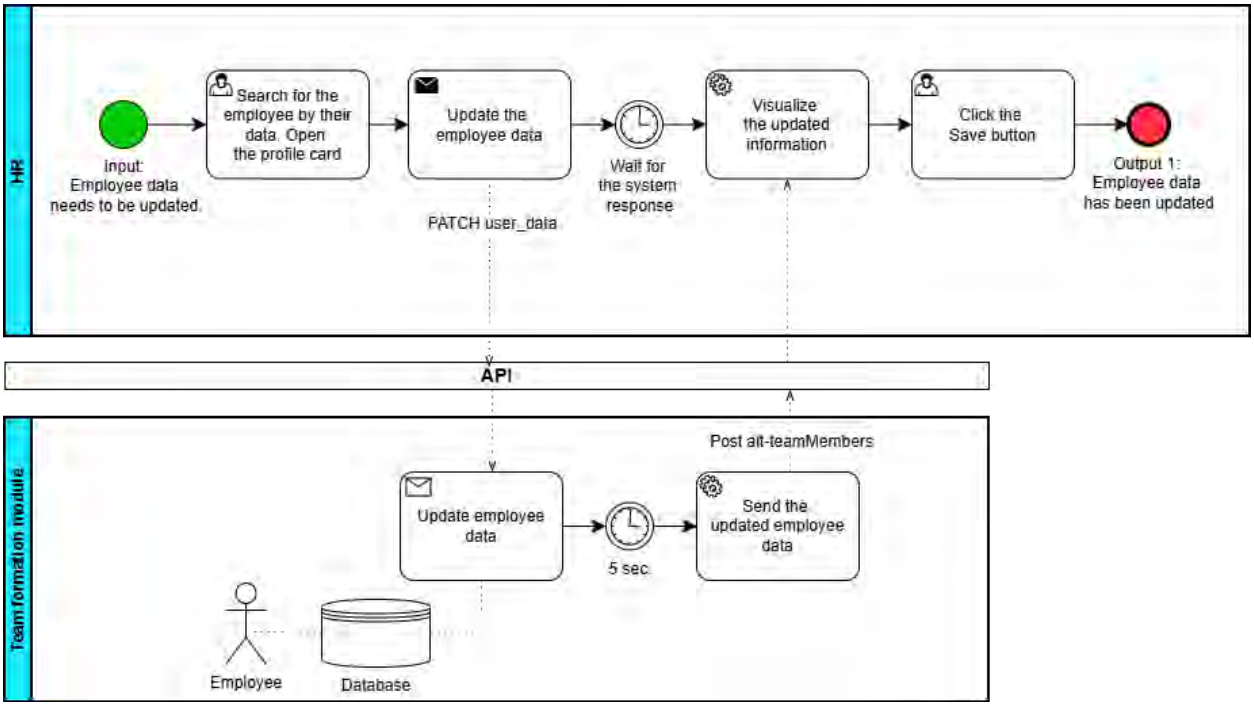

Figure 4. Employee information administration

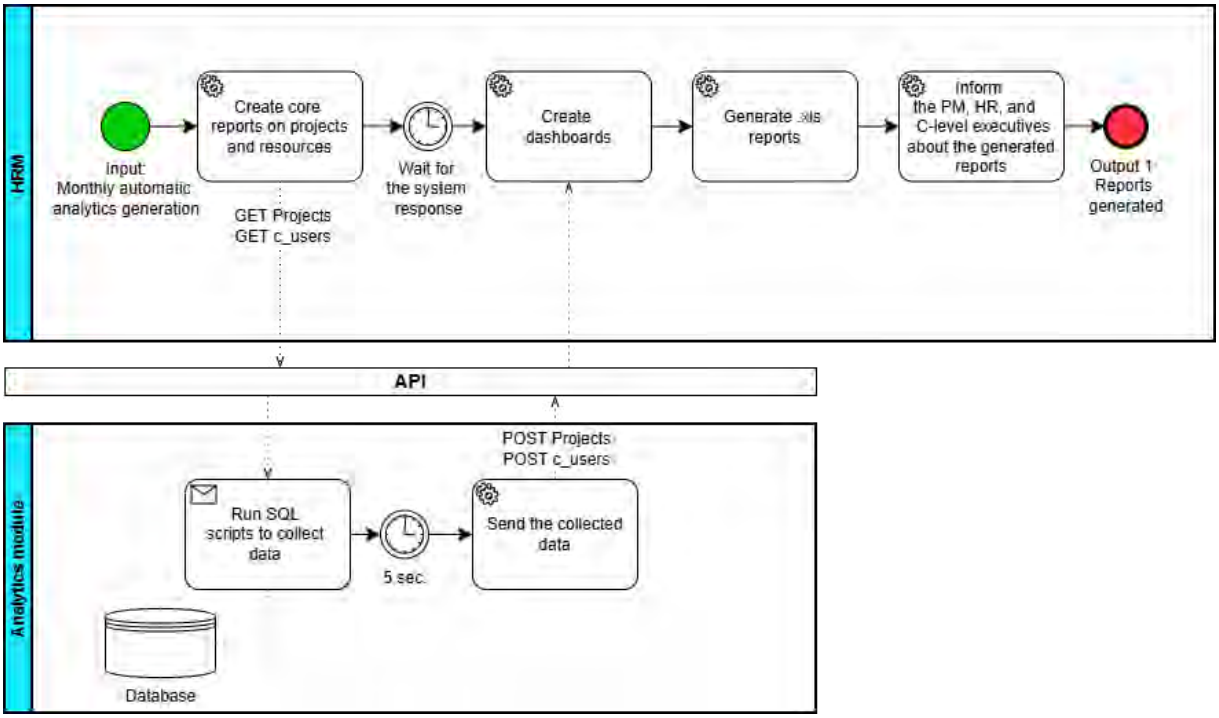

Figure 5. Generation of analytical reports on resource utilization

In study [2], a conceptual model of the HRM system was designed, illustrating the interaction between the database and system services. The data model is aimed at organizing and managing the database structures used for system and team operations.

The primary objective is to identify relationships, define the components of each table, and describe their interactions within the database [3; 4; 5; 6].

The HRM system database consists of the following tables:

1. c_users – contains the list and information of all employees registered in the system.
2. ss_techskills – contains the list of all technical skills of employees.
3. ss_softskills– contains the list of all soft skills and qualitative attributes of employees.
4. c_userskills – contains the mapping of employees to their technical and soft skills.
5. sp_projects – contains the list of all projects and their owners (project managers).
6. sp_projectteams – contains the list of employees assigned to specific projects.
7. c_activitylog – stores information about user interactions with the system.
8. c_whitelistusers – contains a list of users who are not formally employed by the company but have access to its system resources.

The primary entity is employees (users), who may belong to zero or multiple projects. This table also includes external users (c_whitelistusers), who are assigned a specialized type labeled «external». Employees possess up to three dominant technical skills (ss_techskills) and up to three soft skills (ss_softskills), which are taken into account by the employee selection module during team formation. Projects (sp_projects) are associated with a single project team. The activity log (c_activitylog)

contains zero or more records of user actions on the platform. These records store information for a one-month period and are deleted with the same retention frequency.

An employee (1) may belong to one or multiple projects (one-to-many relationship). This entity is illustrated in Fig. 6.

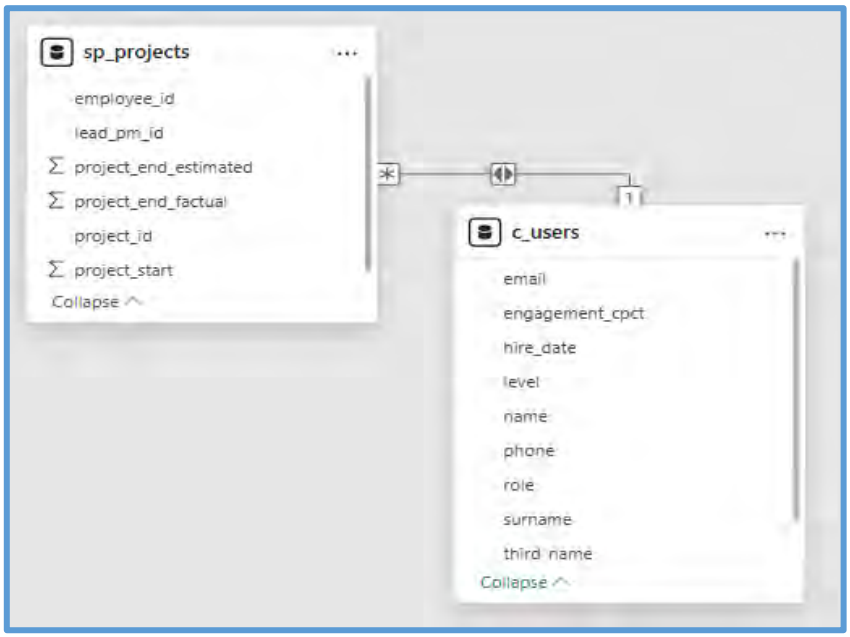

Figure 6. Employee-to-project relationship entity

Employees may also include external individuals, such as sole proprietors (FOPs) or third-party recruiters. Multiple external individuals may be associated with multiple users in the c_users table (many-to-many relationship), as illustrated in Fig. 7.

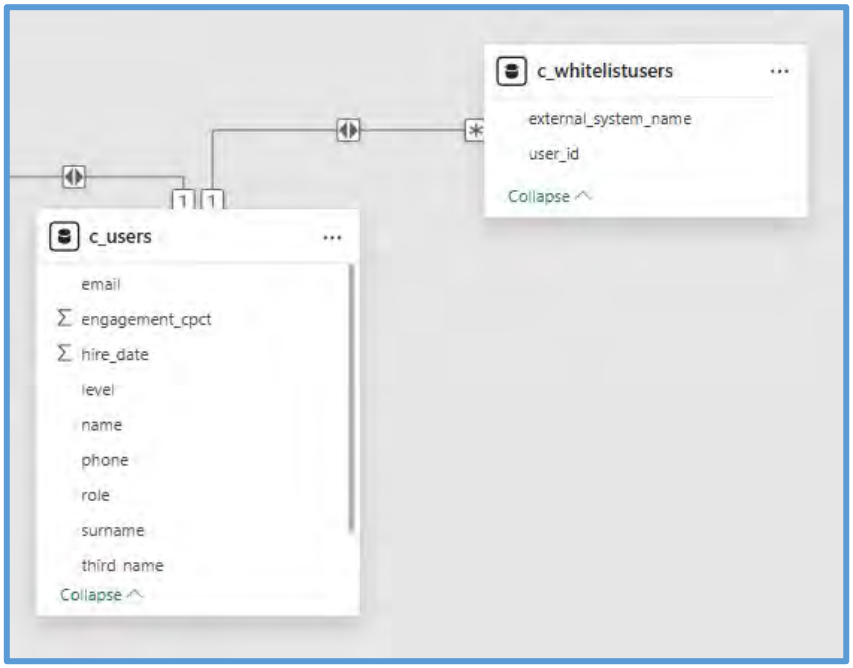

Figure 7. External worker-to-employee relationship entity

The next relationships involve the mapping of employees' technical and soft skills. Each of these tables contains up to three primary user skills, respectively, resulting in a one-to-one relationship. This relationship is illustrated in Fig. 8.

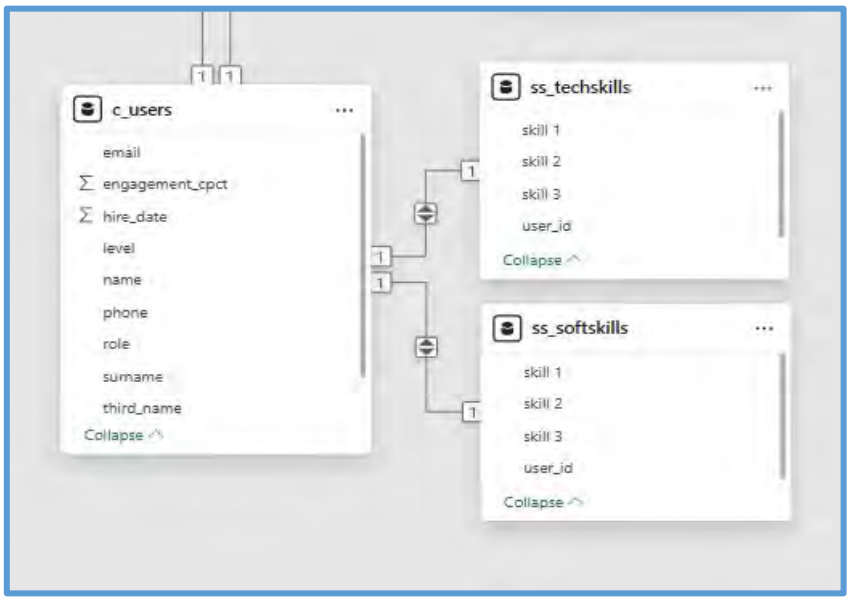

Figure 8. Skill tables-to-employee entity relationship

As mentioned earlier, the platform must log user actions. Accordingly, the c_activitylog table may contain multiple records for a single user (one-to-many relationship). This interaction is illustrated in Fig. 9.

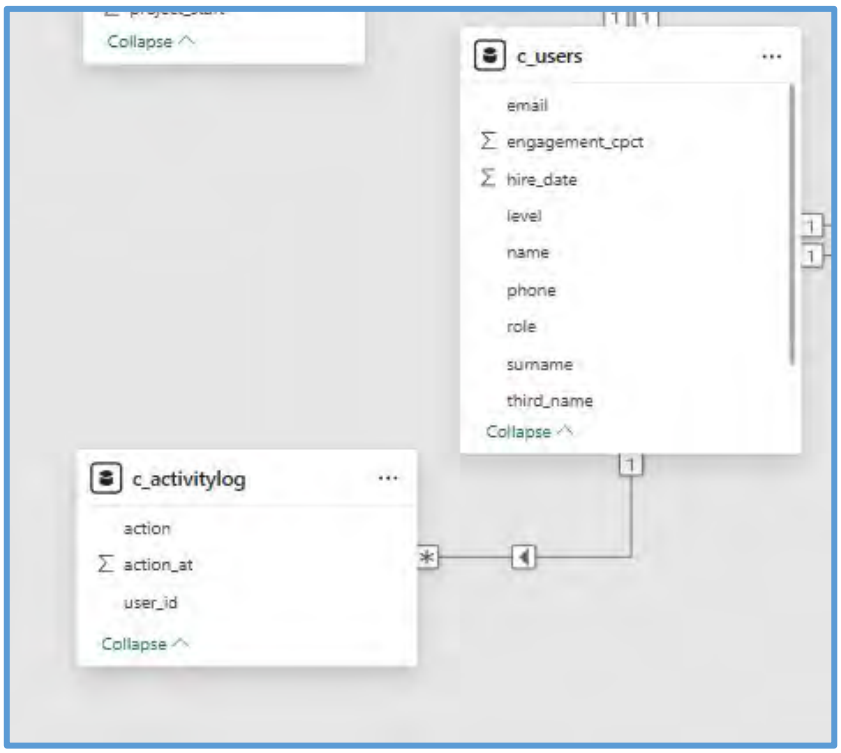

Figure 9. User activity-to-employee entity relationship

The final entity represents the interdependence between the project team, the project itself, and the overall list of employees. Accordingly, teams may be associated with different projects (many-to-many relationship), and team members correspond to the general pool of employees (many-to-many relationship). This interdependency is illustrated in Fig. 10.

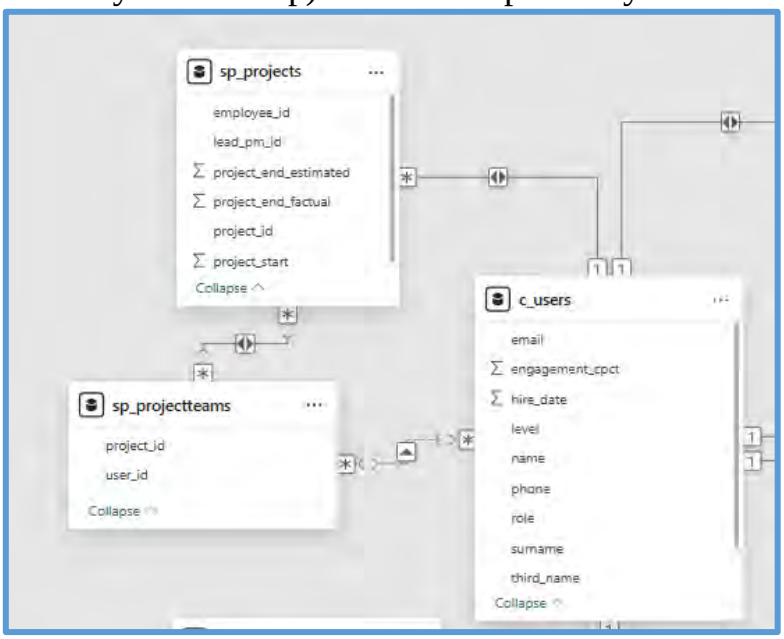

Figure 10. Project team-to-employees and projects entity relationship

The complete visualization of the table relationships is shown in Fig. 11.

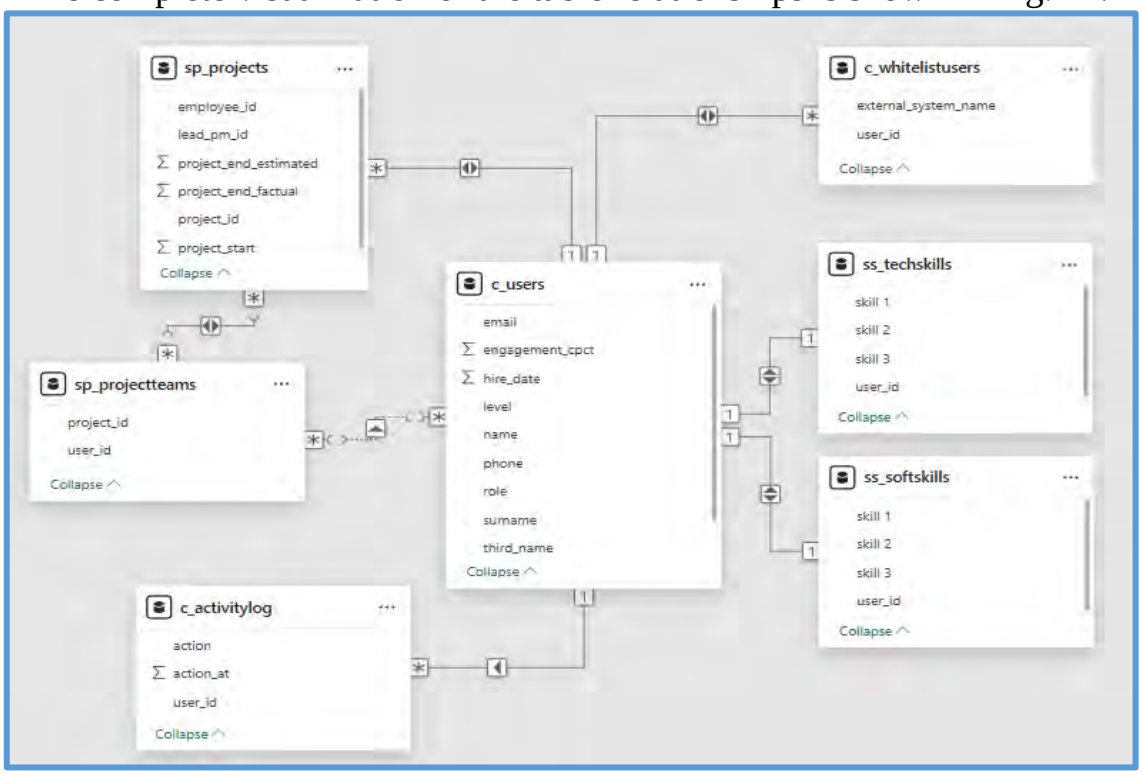

Figure 11. Conceptual database model

After establishing the relationships between the tables, it is necessary to proceed to the development of the data logic model. This model describes the entities, classes,

or data objects relevant to the domain, the attributes used to define them, and the relationships between them, thereby providing a unified semantic framework for analysis and implementation.

The data logic model is a crucial component of modern relational databases [7; 8]. It enables validation of the data structure against normalization rules, eliminates data redundancy, and protects the database from anomalous dependencies related to insertion, update, or deletion operations.

The model clearly defines which tables depend on one another. The resulting logical schema is universal, enabling DevOps teams to deploy the database on any relational platform (PostgreSQL, MySQL, Oracle, or SQL Server [9; 10]) without compromising the system architecture.

Accordingly, it is necessary to define the list of column attributes for the database entities in order to construct the data logical model. The complete list is presented in Tables 1–8.

Table 1. List of attributes of the «c_users» table

Field	Type	Null	Key	Default
user_id	uuid	NO	PRI	NULL
name	varchar(50)	NO	-	NULL
surname	varchar(50)	NO	-	NULL
third_name	varchar(50)	YES	-	NULL
email	varchar(100)	NO	UNI	NULL
phone	varchar(20)	YES	-	NULL
role	varchar(100)	NO	-	NULL
level	varchar(50)	NO	-	NULL
engagement_cpct	decimal(3,2)	NO	-	NULL
hire_date	date	NO	-	NULL

Table 2. List of attributes of the «sp_projects» table

Field	Type	Null	Key	Default
project_id	uuid	NO	PRI	NULL
lead_pm_id	uuid	NO	MUL	NULL
project_start	timestamp	NO	-	NULL
project_end_factual	timestamp	YES	-	NULL
project_end_estimated	timestamp	NO	-	NULL
employee_id	uuid	NO	MUL	NULL

Table 3. List of attributes of the «sp_projectteams» table

Field	Type	Null	Key	Default
project_id	uuid	NO	PRI	NULL
user_id	uuid	NO	PRI	NULL

Table 4. List of attributes of the «ss_techskills» table

Field	Type	Null	Key	Default
user_id	uuid	NO	PRI	NULL
skill 1	varchar(50)	YES	-	NULL
skill 2	varchar(50)	YES	-	NULL
skill 3	varchar(50)	YES	-	NULL

Table 5. List of attributes of the «ss_softskills» table

Field	Type	Null	Key	Default
user_id	uuid	NO	PRI	NULL
skill 1	varchar(50)	YES	-	NULL
skill 2	varchar(50)	YES	-	NULL
skill 3	varchar(50)	YES	-	NULL

Table 6. List of attributes of the «c_activitylog» table

Field	Type	Null	Key	Default
user_id	uuid	NO	MUL	NULL
action_at	timestamp	NO	-	NULL
action	varchar(100)	NO	-	NULL

Table 7. List of attributes of the «c_whitelistusers» table

Field	Type	Null	Key	Default
user_id	uuid	NO	MUL	NULL
external_system_name	varchar(100)	NO	-	NULL

Modern methods and tools enable two main approaches for creating, configuring, and managing databases:

1. UI-based approach. Most modern database management systems provide a visual editor for creating tables, configuring or deleting them, defining relationships between tables, and organizing and modifying data, including row-level operations. This approach is convenient and intuitive for users, as the system automatically generates and executes SQL scripts without requiring direct user involvement. This is especially beneficial for employees who may not have sufficient SQL knowledge or experience in writing database scripts for table creation and management.

2. SQL scripts. This approach requires knowledge of programming language functions and does not provide an extensive user interface for convenience. However, it ensures higher precision and better control over the creation and management of database tables.

Below is an example of an SQL script for creating the c_users table:

```
CREATE TABLE c_users (
    user_id UUID PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    surname VARCHAR(50) NOT NULL,
    third_name VARCHAR(50),
    email VARCHAR(100) NOT NULL UNIQUE,
    phone VARCHAR(20),
    role VARCHAR(100) NOT NULL,
    level VARCHAR(50) NOT NULL,
    engagement_cpct DECIMAL(3,2) NOT NULL,
    hire_date DATE NOT NULL
);
```

An SQL script is composed using the CREATE function, which creates a new record (table) in the database management system. Inside the brackets, the column name

is specified, followed by the attribute type, whether the column is a primary relationship with other tables (PRIMARY KEY), and whether the column value can be non-null.

All unique identifiers in the system use the UUID type because it has a length of 32 characters plus 4 hyphens, which increases the number of possible records and standardizes the approach to identifying database objects.

The PRIMARY KEY indicates that a specific column (for example, user_id) is the unique identifier of each row. Another important aspect of database management is defining relationships between tables. For this purpose, the FOREIGN KEY attribute is used, which defines which columns establish external relationships with other tables. For example, the sp_projectteams table has the following relationship:

```
CREATE TABLE sp_projects (
    project_id VARCHAR(36) PRIMARY KEY,
    lead_pm_id VARCHAR(36) NOT NULL,
    project_start TIMESTAMP NOT NULL,
    project_end_factual TIMESTAMP,
    project_end_estimated TIMESTAMP NOT NULL,
    employee_id VARCHAR(36) NOT NULL,
    -- Definition of foreign keys
    CONSTRAINT fk_projects_pm FOREIGN KEY (lead_pm_id) REFERENCES c_users(user_id),
    CONSTRAINT fk_projects_emp FOREIGN KEY (employee_id) REFERENCES c_users(user_id)
);
```

However, to correctly use a FOREIGN KEY, it is necessary to specify REFERENCES, which defines the relationship to a column in the related table. This is required to establish the link between tables and to enable the creation of joined queries that retrieve data from multiple tables simultaneously.

Conclusion. In the course of this research, an important problem concerning the justification and development of an intelligent HRM system architecture was addressed. The proposed system is based on the concept of «ethical automation» and preventive human capital management in IT projects. The author demonstrates that the transformation of traditional management approaches into measurable economic efficiency can be achieved through the implementation of algorithms that ensure a balance between the technical compatibility of specialists and their psycho-emotional state. In particular, strict adherence to an 85% workload threshold is introduced as a preventive mechanism for burnout reduction. The designed system architecture and the developed database datalogical model enable the automation of multi-criteria team formation processes by integrating hard and soft skills, project roles, and employee personal attributes. The described interaction mechanisms between the web platform, API, and the employee selection module ensure high accuracy in specialist matching and enable the rapid formation of alternative team compositions. This represents a strategic factor in ensuring the long-term competitiveness of IT businesses in highly volatile market conditions.

References

1. Ziuziun V., Kulkovets V., & Parasiuk L. (2024). Development of a Decision Support Information System for Managing Large Agile Teams in IT Projects. Collection of Scientific Papers of Admiral Makarov National University of Shipbuilding, 4(497), P. 166–172. https://doi.org/10.15589/znp2024.4(497).23
2. Ziuziun, V., & Korytnyi, O. (2026). Intelligent Model for Ranking IT Specialists Based on Competencies, Team Roles, and Resource Workload Indicators. Proceedings of the 4th International Scientific and Practical Conference «Modern Scientific Research: Theoretical and Practical Aspects», P. 246-257. https://doi.org/10.70286/EOSS-11.05.2026.009.246-257
3. Torres-Jimenez, J., Rangel-Valdez, N., De-la-Torre, M., & Avila-George, H. (2022). An Approach to Aid Decision-Making by Solving Complex Optimization Problems Using SQL Queries. Applied Sciences, 12. MDPI. https://doi.org/10.3390/app12094569
4. Katsogiannis-Meimarakis, & G., Koutrika, G. (2023). A survey on deep learning approaches for text-to-SQL. The VLDB Journal, 32. Springer, Berlin. https://doi.org/10.1007/s00778-022-00776-8
5. Eido, W.M., & Yasin, H.M. (2025). Machine Learning Approaches for Enhancing Query Optimization in Large Databases. Engineering And Technology Journal, 10. https://doi.org/10.47191/etj/v10i03.39
6. Codd, E.F. (1970). A Relational Model of Data for Large Shared Data Banks. Communications of the ACM, 1970, 13(6), P. 377–387. https://doi.org/10.1145/362384.362685
7. Ng, P.A. (1971). Further Analysis of the Entity-Relationship Approach to Database Design. IEEE Transactions on Software Engineering, 7(1), P. 85–99. https://doi.org/10.1109/TSE.1981.234522
8. Al-Fedaghi, S. (2024). An Enhanced Framework for Object-Relational Database Design – An Experimental Study. 2024 IEEE International Conference on Database Technologies and Applications (ICDTA), P. 105–112. https://doi.org/10.1109/icdta.2024.10808665
9. Ziuziun, V., & Demchuk, G. (2025). Information Architecture and UI/UX Design of a Mobile Application for Coworking Spaces. Proceedings of the 3rd International Scientific and Practical Conference «Innovations in Science: From Theoretical Foundations to Practical Impact», P. 102-111. https://doi.org/10.70286/EOSS-24.11.2025.002
10. Ziuziun, V., & Bredikhin, D. (2025). Development of the Information Structure of the Project for Creating a Web Platform to Improve the Ecological Culture of the Population. Collection of Scientific Papers of Admiral Makarov National University of Shipbuilding, 1(499), P. 126-136. https://doi.org/10.15589/znp2025.1(499).18