 [Translated from Ukrainian to English - www.onlinedoctranslator.com](https://www.onlinedoctranslator.com/en/?utm_source=onlinedoctranslator&utm_medium=docx&utm_campaign=attribution)

CONTENTS

[List of symbols and abbreviations 4](#_Toc169612913)

[Introduction 5](#_Toc169612914)

[1 Organization of planning of the training load with assessment of distribution parameters 7](#_Toc169612915)

[1.1 Description of the workload planning process 7](#_Toc169612916)

[1.2 Organization of initial load distribution with assessment of its quality 13](#_Toc169612917)

[2 Development of algorithmic and information support for the process of assessing the distribution of the educational load 17](#_Toc169612918)

[2.1 Development of algorithms for assessing the distribution of the educational load 17](#_Toc169612919)

[2.1.1 Development of a model for assessing the completeness of the distribution 17](#_Toc169612920)

[2.1.2 Development of a model for assessing the correctness of the distribution 19](#_Toc169612921)

[2.1.3 Development of a distribution quality assessment model 20](#_Toc169612922)

[2.2 Development of mathematical support for calculating indicators of the distribution of educational workload 24](#_Toc169612923)

[2.3 Organization of information storage for calculating indicators 30](#_Toc169612924)

[2.3.1 General information about databases and database management systems 30](#_Toc169612925)

[2.3.2 Development of a database structure to support calculations 32](#_Toc169612926)

[2.3.3 Choosing a database management system format 36](#_Toc169612927)

[3 Designing a program for calculating workload distribution indicators 38](#_Toc169612928)

[3.1 Developing a task to create a software solution 38](#_Toc169612929)

[3.1.1 Product identification 38](#_Toc169612930)

[3.1.2 Glossary 39](#_Toc169612931)

[3.1.3 Software solution requirements 39](#_Toc169612932)

[3.2 Defining the technology stack for development 42](#_Toc169612933)

[3.3 Testing organization 46](#_Toc169612934)

[4 Description of preparation for application and use of the software solution 50](#_Toc169612935)

[4.1 Description of preparing the software solution for use 50](#_Toc169612936)

[4.2 Description of the procedure for using the software functionality to generate statistics 53](#_Toc169612937)

[5 Economic justification of the software development project 59](#_Toc169612938)

[5.1 Justification of the feasibility of software development 59](#_Toc169612939)

[5.2 Assessment of software competitiveness in comparison with its analogue 59](#_Toc169612940)

[5.3 Justification of the feasibility of software development 60](#_Toc169612941)

[5.4 Calculating project costs for software development 64](#_Toc169612942)

[5.5 Calculating the cost of software implementation 65](#_Toc169612943)

[5.6 Calculating the cost of software implementation 66](#_Toc169612944)

[5.7 Calculation of current operating costs of software and its analogue 66](#_Toc169612945)

[5.8 Calculation of the economic effect of software development 69](#_Toc169612946)

[Conclusions 71](#_Toc169612947)

[List of information sources 73](#_Toc169612948)

# abbreviations

DB – database;

BP – business process;

DAK – State Attestation Commission;

Higher education institution – institution of higher education;

IZ STS – individual lesson on the topic of independent work;

LS – laboratory class;

NPP – scientific and pedagogical worker;

NTU "KhPI" - National Technical University "Kharkiv Polytechnic Institute";

PrZ – practical classes;

PIITU – software engineering and intelligent management technologies;

Software – software;

RNN – distribution of the loading load

DBMS – database management system.

# Introduction

Every year, the department is faced with the need to calculate the teaching load for the teaching staff. The results of such planning determine the content of the activities of all teachers for the current academic year. The result of this process is a plan for assigning teachers to conduct lectures and practical classes for the current semester. An important issue is the assessment of the completeness and quality of the load distribution. Effective load distribution ensures high quality of education, which is one of the tasks of the department. Therefore, the organization of the assessment of the distribution of the teaching load is an urgent task. The peculiarity of the assessment is the need to take into account the results in a very short time. This determines the need for the calculation of the parameters necessary for evaluating the obtained variant of the distribution of teachers, which is planned to be solved in this work, to be carried out automatically.

The object of the study is the methodology for planning the distribution of the educational workload of the department's scientific and pedagogical staff at the beginning of the academic year.

The subject of the study is a model for representing the distribution of the teaching workload of the department's scientific and pedagogical staff, indicators for evaluating the distribution, and models for their calculation.

The purpose of the work is to automate the calculation of indicators of completeness, correctness, and quality of distribution of the teaching workload of the department's scientific and pedagogical staff in order to reduce the time required to perform this operation compared to manual calculations.

The task of the work is to develop algorithms for calculating a certain set of indicators of the distribution of the educational load, to solve the issue of presenting the necessary data, to develop software that implements calculations in accordance with the algorithms, and also presents them in a convenient form.

The sections of the main part of the work consider the construction of a business process for distributing the educational workload, the definition of the basic concepts of the subject area of ​​​​this task, the development of mathematical, information and software for performing the relevant calculations, as well as the economic justification for creating the program.

The features of the organization of the distribution of the teaching workload of scientific and pedagogical workers at the beginning of the new year and the content of the proposed assessment approach are considered in the first section.

# 1 Organization of planning of the training load with assessment of distribution parameters

## 1.1 Description of the workload planning process

In the modern world, the management of educational activities in universities requires the coordination of a large number of specialists from different fields. This can be achieved through the organization of reporting on various types of activities, holding organizational and methodological meetings of departments, conducting scientific and methodological seminars to improve the scientific level and the level of methodological readiness for conducting various types of classes, etc. Since NTU "KhPI" is a state institution subordinate to the Ministry of Education and Science of Ukraine, this process is complicated by the need to comply with numerous laws, regulatory acts and orders, in particular, the Law of Ukraine "On Higher Education" [1].

Calculating the workload requires taking into account numerous factors that often change and have limitations. A change in any of these factors requires that the calculations be revised. The reason for the changes may be the introduction of a new order of the Ministry of Education and Science of Ukraine, a revision of curricula, or a restructuring of university departments.

Typically, such changes are implemented and approved at the end or beginning of a new academic year. However, there are situations when changes occur during an academic semester or between semesters. Therefore, calculations of the teaching load and preparation of relevant documents should not only be regular and planned, but also require prompt updating in a limited time. This increases the risk of errors due to the human factor, since data processing requires a focus on numerical data and the relationships between them, subjects and teachers.

Software development usually begins with studying the needs of future users and determining how they perform the functional duties that are planned to be automated [2]. This process is called business process modeling. In DSTU, the definition of a business process is given in the ISO 9000:2015 standard. According to it, a business process (BP) is “… a set of interrelated or interacting activities that transform inputs into outputs, creating value for the consumer, whether internal or external. It is a sequence of cross-functional activities that has a clearly defined beginning and end, and that is repeated in time” [3]. A business process has clearly defined components: a goal, input and output data, as well as performers. The theory of BP modeling and refactoring can be considered in [4].

The title of the BP under consideration is "Formation of the distribution of the teaching load (TLD) of scientific and pedagogical workers (STP) for the academic year."

The unit where the business process is implemented: the educational part of the PPITU department.

Performer: head of the department and employees involved in planning the educational process.

Initial data for development:

* curriculum for the current academic year;
* data on academic groups;
* data on the curricula followed by students in the groups;
* data on blocks of elective subjects chosen for study by students of academic groups;
* data on postgraduate students assigned to the department;
* department staffing list.

At the PIITU department, such a calculation is carried out in an Excel workbook. Sheets with the formed distribution (for the fall and spring semesters) contain:

* lines corresponding to the lecture schedule, which indicate the number of hours for their reading, as well as for consultations and for conducting a final control: either a test or an exam;
* lines corresponding to the planning of practical or laboratory classes, which indicate the number of study groups, the number of hours per group, as well as the total number of hours;
* lines containing data on the management of course and diploma projects (theses), training of postgraduate students, participation in the reception of course project (theses) defense;
* lines reflecting hours of work as part of the State Attestation Commission (SAC), including reviewing papers.

An example of the distribution of the teaching load, created at the beginning of the current academic year, is presented in Fig. 1.1.

The stages of preparing for the distribution of the teaching load are as follows:

1. receiving curricula and a list of student groups in electronic form;
2. formation of a basic RNN for NPP, which includes a list of disciplines, distribution of hours by type of classes, type of reporting and semester of study;
3. creating streams from academic groups in accordance with the vision of management, adhering to the requirements of guiding documents;
4. formation of RNN NPP terms in accordance with the list of flows;
5. derivation of formulas, calculations and registration of RNN of NPP in the established form;
6. assigning lecturers to educational streams, appointing teachers to conduct practical classes;
7. appointment of supervisors of diploma and course design.
8. appointment of SAK members and reviewers.

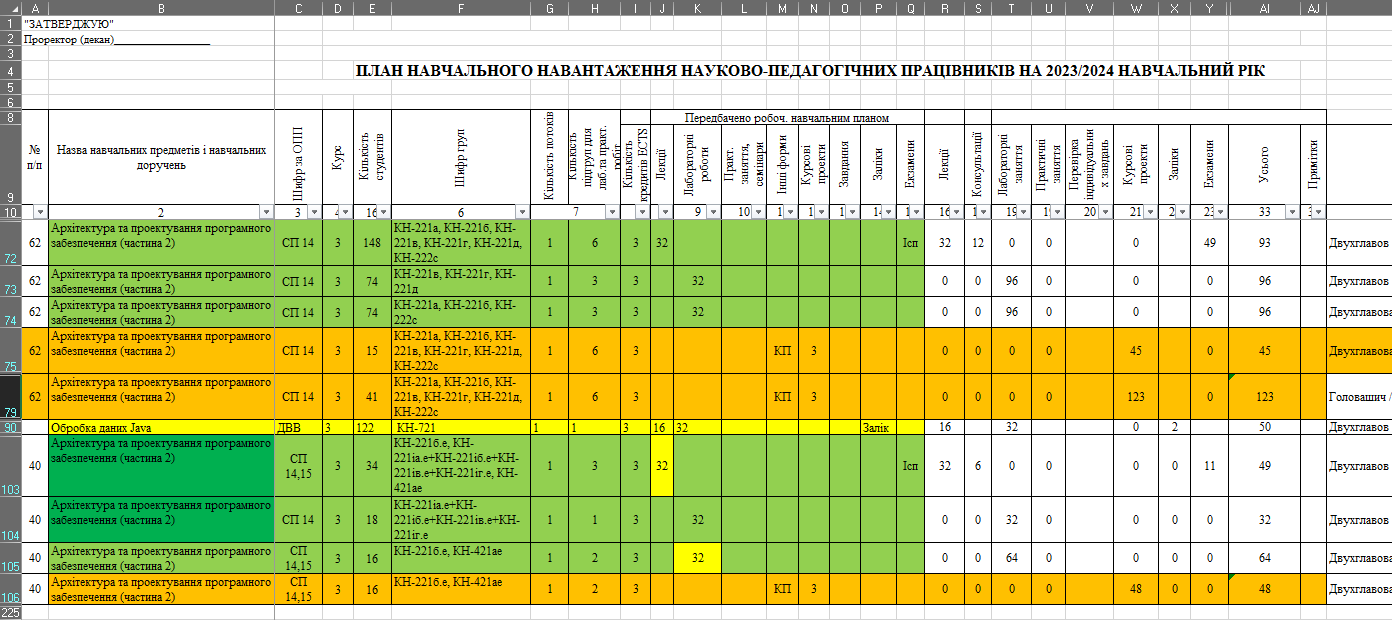


Figure 1.1 – Example of an Excel sheet with a fragment of the distribution of the teaching load of the PIITU department

Developing such a plan is a creative work that requires some experience. The general stages of this work were described in the article [5], as well as  
in his master's thesis [6], dedicated to the creation of a support system for the distribution of the study load "Studyload docs" (SLDOCS).

The SLDOCS system was created to automate the process of compiling a teaching report, its coordination, detection and correction of errors. This is a multi-iterative process that includes monotonous manual operations related to the preparation of extracts for teachers. According to [6], SLDOCS is intended for automated creation of information on teaching assignments, an individual teaching report for an individual teacher and a template for an individual teacher's work plan for the academic year. The input data are teaching reports in Excel workbook format and a list of teachers with their email addresses. The system allows you to download or send documents in XLSX format to email, and also allows teachers to independently create accounts and download documents after gaining access.

After completing the creative work on creating a training load plan, the file in Excel workbook format is loaded into the corresponding tables of the system database via the interface. The appearance of the data after loading can be seen in Fig. 1.2. This is the result of a query that combines several tables for clarity. Convenient data presentation can be provided using the Spring framework [7] and the ThemeLeаf templating tool [8], which were used in the development of “SLDOCS”. Fig. 1.3 presents the logical diagram of the database in MySQL8 format [9] for generating extracts from RNN.

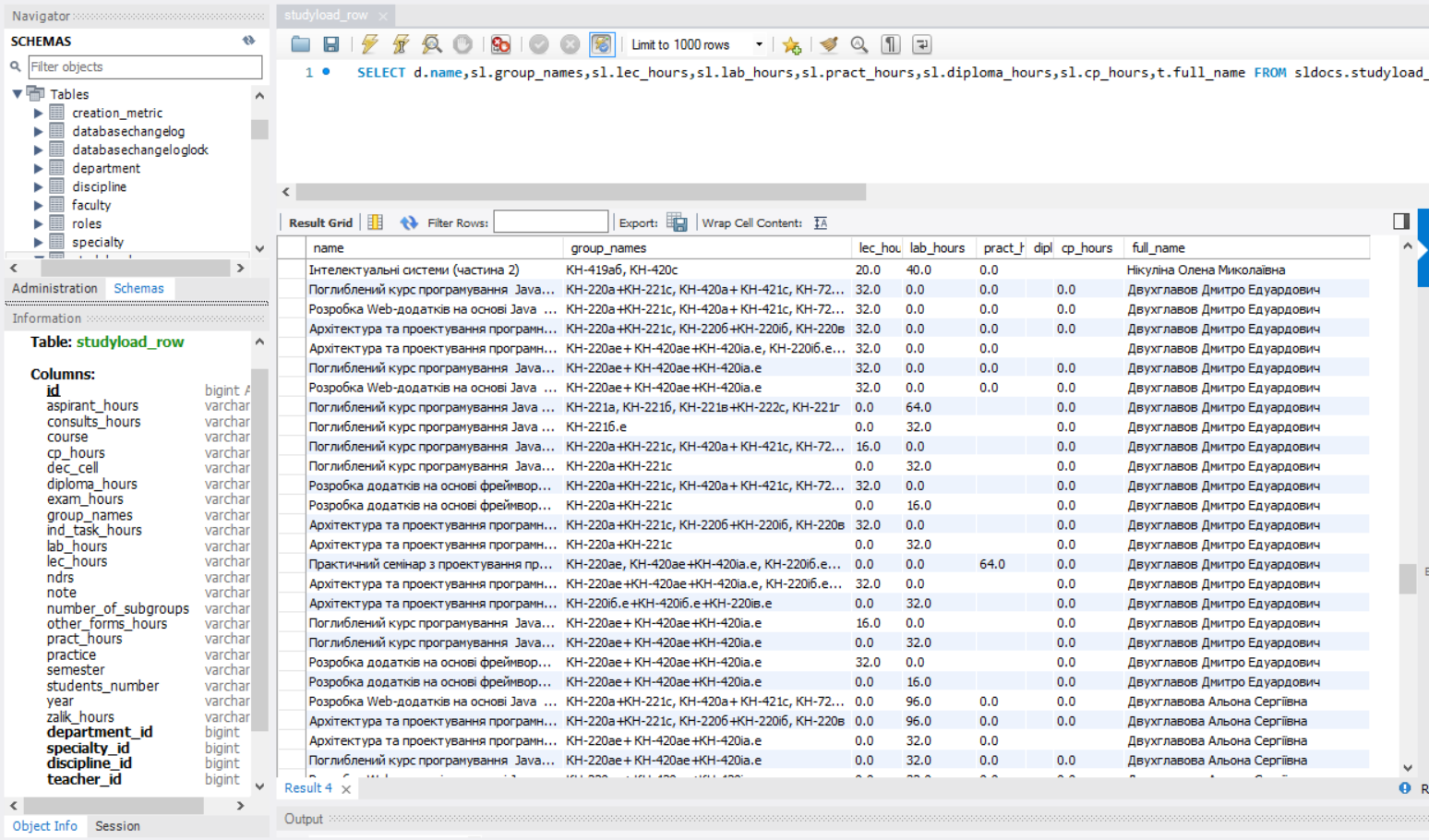


Figure 1.2 – Filling the SLDOCS database after loading information on the distribution of the teaching load from an Excel file

The implementation of the system significantly reduces the time for delivering the workload to teachers. The presence of the RNN stored in the database (DB) allows you to automate the solution of other tasks. Participants in the process of planning the workload have the opportunity to automatically control the distribution for errors, as well as prepare various summary documents that allow you to see the general picture of the formed distribution. This is important, since a teacher must have a workload of 600 hours per rate per academic year. If the number of hours is higher or lower than 600 hours (for one rate) by more than 20%, then a redistribution of the workload must occur. The load norms are defined in [10].

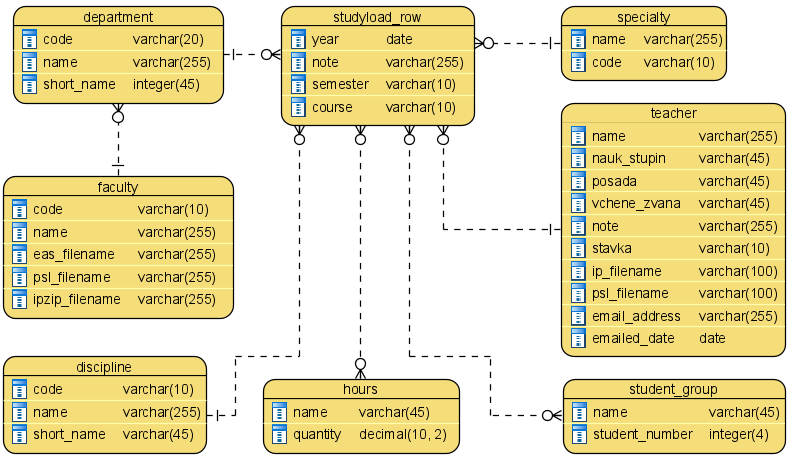


Figure 1.3 – Diagram of the “SLDOCS” software database

Further considerations showed that the presence of the NPP PNN loaded into the database opens up wide opportunities for their processing. . If you look at the data stored in the database, you can see that it stores in a linked form information about teachers, their positions, academic degrees (Candidate of Sciences or Doctor of Sciences) and academic ranks (professor, associate professor, academician, senior researcher), as well as information about disciplines, which can be selected by types of educational workload (lectures, laboratory or practical classes, supervision of course projects (works), diploma design of masters or bachelors). In particular, it is possible to express generalized data by distribution, for example, the number of lectures given by doctors of sciences and senior teachers, or to calculate the share of lectures in the total number of hours for each associate professor. The capabilities of many programming languages ​​​​also allow them to be presented in a form convenient for analysis - tables, histograms, diagrams, graphs. The synthesis of all previous ideas regarding the use of digital storage of PNN led to the creation of the business process “PNN Development with Quality Assessment”, which is discussed in the next section. It should be considered as a proposal for improving the existing development process.

## 1.2 Organization of initial load distribution with assessment of its quality

The business process model proposed for implementation is presented in Fig. 1.4.

It is proposed to expand the existing BP with an activity that, after creating the initial version of the distribution plan, will be followed by the calculation of statistics and its evaluation.

It is planned that the developer of the RNN plan will actively use the existing "SLDOCS" system. At the same time, the functionality of the system should be expanded by adding new functions, in particular, a set of classes and methods for calculating the characteristics of the formed version of the RNN.

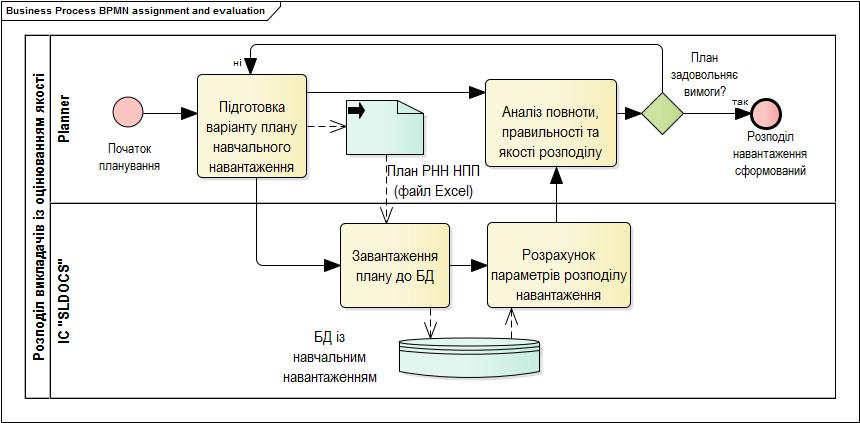


Figure 1.4 – Model of training load distribution with quality assessment

The business process model is presented using BPMN (Business Process Model and Notation) – a notation for modeling business processes that stores a description in an XML file, providing the ability to export/import both the entire model and its individual packages. It was developed by the Business Process Management Initiative and is supported by the Object Management Group after the merger of both organizations in 2005. The latest version of BPMN is 2.0.1 [11]. Enterprise Architect is one, but the main, program that allows you to create diagrams of this type. It is claimed that it is “… an ideal solution for organizations and enterprises in the field of visualization, analysis, modeling, testing and maintenance of their structure, software, processes and architecture. It is a platform that provides the ability to control the workspace, support colleagues and the team, promotes collaboration and builds confidence in the most complex projects” [12].

Analyzing the business process diagram in Fig. 1.4, we can see that the analysis of the NPP is planned to be carried out in three directions:

* completeness of distribution;
* correctness of distribution;
* quality of distribution.

Completeness of the distribution of the educational load is a characteristic of the distribution option that determines the provision by the teacher of all classes planned for holding. To determine the characteristic, it is necessary to review all items of the list of educational assignments, compare them with the RNN and check:

* defining a lecturer for all lecture streams;
* identification of teachers for all groups created to conduct laboratory work and practical classes;
* appointment of supervisors of qualifying theses (coursework, bachelor's or diploma) for all students.

The correctness of the distribution of the teaching load is a characteristic of the distribution option that determines the assignment of the volume of teaching load determined by the norms to all full-time teachers of the department.

The quality of the distribution of the teaching load is a characteristic of the distribution option that determines the suitability of assigning teachers to conduct certain types of classes in accordance with their categories, which are specified in the methodological guidelines of the university and the department.

The indicators of this group can be divided into two categories:

Individual distribution quality indicators:

* number of hours and percentage for lectures and other related activities (consultations, taking exams);
* number of hours and percentage for practical classes (laboratory work and practical classes);
* number of hours and percentage for course design guidance;
* number of hours and percentage for supervising bachelor's and/or master's thesis design.

Indicators of the quality of the distribution of classes in the subdivision:

* number of hours and percentage of classes conducted by doctors of sciences and professors;
* number of hours and percentage of classes conducted by candidates of sciences, associate professors;
* number of hours and percentage of classes conducted by NPP without ranks and degrees.

A thorough description obtained during the analysis of the business process allows you to identify the tasks that need to be solved for its implementation. These stages should include:

* formation of a set of indicators for evaluating the RNN variant;
* formation of mathematical formulas for calculating the indicators of the evaluation of the RNN variant, as well as algorithms for their calculation;
* develop a data storage model for calculations, as well as for storing indicators of distribution options;
* selected technology stack, design, develop and test software for calculating distribution option indicators;
* prepare demonstration examples of the software's operation, as well as qualitatively analyze the correct functioning of the software solution that will be created.

# 2 Development of algorithmic and information support for the process of assessing the distribution of the educational load

## 2.1 Development of algorithms for assessing the distribution of the educational load

### 2.1.1 Development of a model for assessing the completeness of the distribution

Completeness of the distribution of the teaching load is a characteristic of the distribution option that determines the teacher's provision of all classes scheduled for delivery.

Currently, such a curriculum distribution indicator (CDI) can be determined more quickly using the capabilities of the Microsoft Excel spreadsheet. To do this, you can simply set a filter on the column with the names of teachers, and then select the filter option “Empty rows”. On the other hand, if the “Empty rows” option is not in the list, this indicates that the load distribution is carried out in full [13].

The absence of managers with master's or bachelor's degrees is determined similarly. You can also apply a filter and get the result.

But there is one point that determines the need for changes - now the distribution for autumn and spring are located on different sheets. This requires setting filters twice. Fig. 2.1 shows a possible option for presenting the results of the stage execution, when the data is summarized on one sheet after simply clicking a button.

Fig.2.2. shows a variant of presenting data about diploma students and their supervisors. Here, in one place, information is summarized about all categories of students who write qualification papers. In other words, all students of 4th, 5th and 6th years.

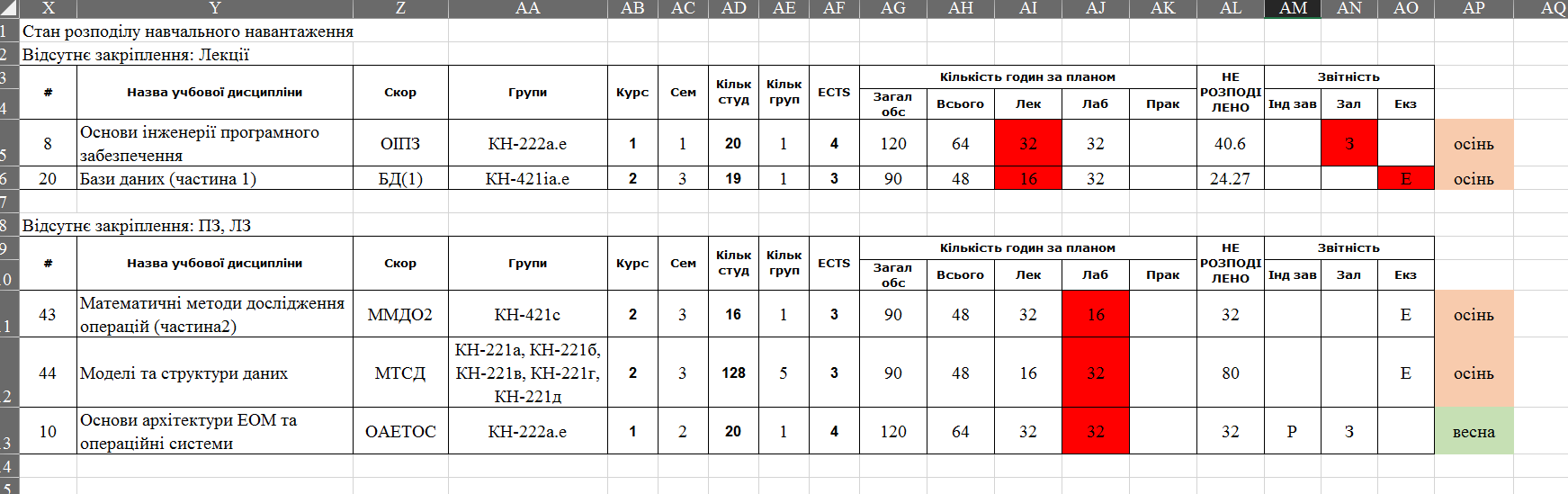


Figure 2.1 – Presentation of information for assessing the incompleteness of teacher assignments to disciplines

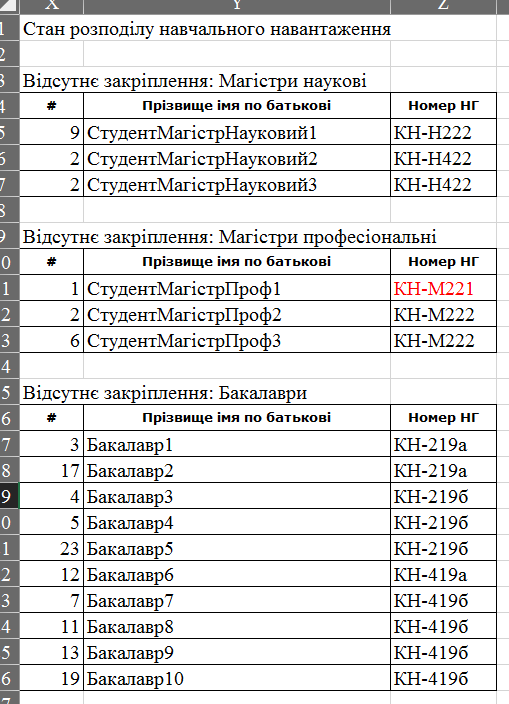


Figure 2.2 – Presentation of information for assessing the incompleteness of the appointment of thesis project supervisors

Both options demonstrate an option that allows you to save all identified deficiencies in a separate file. This file can then be printed, mailed to the head of the department for analysis, or presented on a presentation slide that will be shown at a staff meeting. Also, a list in the format of “all students without 4th and 5th year supervisors” can be placed on a general display for teachers to review when selecting new students in the near future.

The next step is to evaluate the correctness of the distribution.

### 2.1.2 Development of a model for assessing the correctness of the distribution

The correctness of the distribution of the teaching load is a characteristic of the distribution option that determines the assignment of the volume of teaching load determined by the norms to all full-time teachers of the department.

Automation of calculations of indicators of this group is a more resource-intensive task, and therefore more relevant. An example of a page with summarized data on the workload of teachers is presented in Fig. 2.3.

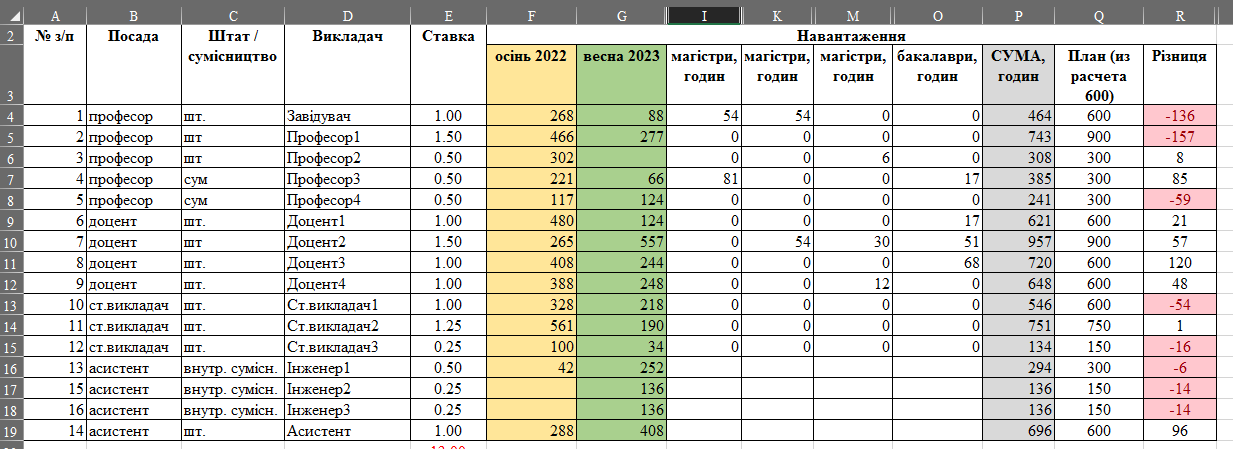


Figure 2.3 – Presentation of information for assessing the quality of workload distribution

Now, to calculate such indicators, it is necessary to transfer from the individual teacher's workload sheet information about the total number of hours of workload for the spring and fall, and then add data about bachelor's and master's degrees. For a department with more than 50 teachers, this task is solved from 1 to 3 hours.

In addition, it is necessary to take into account that the detection of incorrect assignment of hours will lead to the reworking of the plan, which will entail new coordination of changes, recalculation of data, and re-filling of the summary table. That is, it will be necessary to spend another 1 to 3 hours of time.

Calculations of the correctness indicator should be based on the norms of hours of teaching workload, determined by the governing documents of the institution. The normative number of hours of teaching work per one teacher's rate is 600 hours. In this case, the number of hours may differ in both directions by no more than 20% [10].

Additionally, the department may take into account additional indicators at this stage:

* – the minimum number of bachelor's degrees assigned to a diploma for associate professors or master's degrees for professors,
* the minimum number of hours taught in English with a certificate,
* etc.

In this work, the calculation of these indicators is not considered.

### 2.1.3 Development of a distribution quality assessment model

The quality of the distribution of the teaching load is a characteristic of the distribution option that determines the suitability of assigning teachers to conduct certain types of classes in accordance with their categories, which are specified in the methodological guidelines of the university and the department.

Analysis of this distribution direction will allow determining the rationality of assigning teachers to conduct classes defined by the curricula. Scientific and pedagogical workers conduct various types of classes - lectures, laboratory classes (LC), practical classes (PrZ). They manage the implementation of coursework (CR) or course projects (CP), as well as diploma theses (CP). Each of these types of work requires different efforts.

If we consider a situation where one teacher gives 32 hours of lectures, and another conducts 64 hours of laboratory classes, this does not mean that the practical teacher has worked twice as much. First, 64 hours of lectures can be 32 hours of classes in two groups of the stream. Secondly, very often the basic part for methodological developments for conducting lectures or practical classes is prepared by the lecturer, and this is much more than the 32 hours that the practical teacher supposedly worked on. And you should also add to the number of “overtime hours” for solving organizational issues, conducting consultations and exams, developing syllabi, lectures and presentations, and as a result, a completely different calculation result will be obtained.

That is why there is a “hierarchy” of the teaching world, which determines that lectures should be conducted by experienced teachers who hold the positions of associate professors and professors, but the number of disciplines in which they are stream lecturers should not exceed 2-3 disciplines. On the other hand, associate professors who perform the work of assistants, conducting a bunch of lectures and practical classes, but are not responsible for reading lectures or do not work with graduate students, create an additional burden on another colleague and do not engage in self-development and improving their professional level. Therefore, it is necessary to assess the quality of distribution in order to identify “skewnesses” when assigning NPP to conduct classes.

To assess the quality of the distribution, it is necessary to extract the final data of each teacher's individual distribution and calculate the sum of the values ​​of certain columns of classes, depending on the type of activity.

To calculate the number of hours for lecture work, you need to find the sum of hours for lectures, consultations, evaluation of individual tasks within the hours of independent work, as well as for tests and exams.

To calculate the number of hours for practical classes, you need to find the sum of time for conducting PR and PL.

To calculate the number of hours for course design, you should find the amount of time spent advising students on the issue of completing and completing a coursework or project in which the teacher is appointed as the supervisor. This number does not include the hours during which the supervisor of the graduate student provides advice on completing a coursework on a topic related to the subject of the DP. However, the number of hours for this section of activity includes hours spent advising the graduate student, as well as preparing an intermediate coursework (or several works as in the 5th year of the master's degree).

The set of "other types" includes work in the State Attestation Commission (SAC), reviewing bachelor's theses, and managing various types of internships (production, computing, pre-diploma).

The set of “sections” for evaluation can be expanded, which should be taken into account when building the software.

The result of such a calculation may look like the one shown in Fig. 2.4.

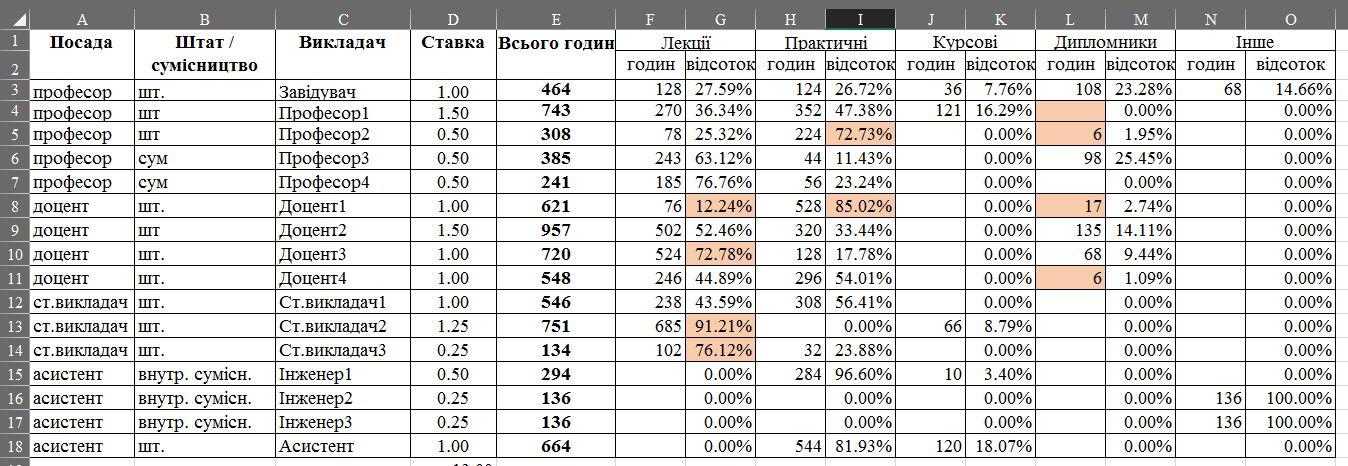


Figure 2.4 – Letter with summary data on RNN

If we calculate the characteristics described above, it will make it possible to assess the distribution of the teacher's time for conducting classes of different types. The availability of such a table will provide the head of the department with the opportunity to analyze the current distribution in order to identify situations where lectures are given by senior teachers without degrees or, conversely, a professor conducts laboratory classes. In Figure 2.4, we can see a situation where the course design is managed by an ordinary assistant. Will this ensure the proper quality of this important stage of training?

In addition to analyzing individual indicators of the conduct of classes by teachers, it is also possible to assess the quality of the organization of classes by type. The department must meet certain licensing and accreditation requirements [10], for example, that classes are conducted exclusively by teachers with a scientific degree. To determine whether the distribution satisfies these conditions, it is advisable to build a “portrait of the department” in the form presented in Fig. 2.5. The figure shows the data on the number of hours planned for conducting previously defined types of classes by three categories of teachers:

* doctors of sciences or professors;
* candidates of sciences or associate professors;
* teachers without academic degrees or academic titles.

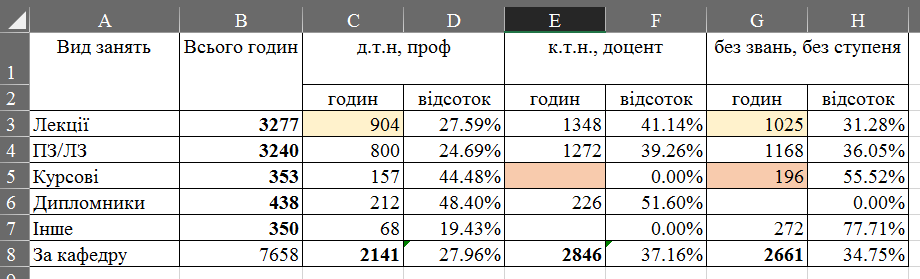


Figure 2.5 – Letter on the distribution of teaching load with data on academic titles and degrees of teachers

Analysis of the statistic given in the table makes it possible to assess the distribution. In addition, the presence of such a statistic allows us to see a more “interesting” picture for the current distribution – associate professors have no relation to course design at all, but at the same time conduct a greater number of practical and laboratory classes than teachers without degrees. This is not logical from the point of view of the functional responsibilities of this category of NPP positions.

Similar tables can be constructed to determine the presence of a foreign language certificate at level B2 for teachers who conduct lectures and practical classes with foreign students in English. It is also possible to add additional requirements for professional skills and certificates for teachers who lecture or conduct classes, for example, in program development or database design. This will ensure a high level of practical training for students.

To build algorithms, one must have either a mathematical apparatus for calculating the indicator or a solution scheme for the problem. The formalization of the calculation of distribution quality indicators is carried out in the next subsection.

## 2.2 Development of mathematical support for calculating indicators of the distribution of educational workload

The correctness of the distribution of teachers can be formally described by the following expression:

;

,

where – is the indicator of correct load distribution;

– indicator of the correctness of the load distribution of the i-th teacher;

– number of teachers;

– the number of hours scheduled for the i-th teacher;

– the standard number of hours for the i-th teacher;

– the coefficient of the rate allocated to the teacher.

Thus, this indicator, with proper distribution, should be equal to one. If this indicator is greater than 1, then this indicates that the teacher has too much workload, if less, then there is a shortage.

To calculate the values ​​of the completeness indicators of the quality of the RNN, it is necessary to organize the extraction of data on the distribution. The technical documentation for the SLDOCS system [6] states that the distribution data is stored in a relational database (DB). This work also presents logical expressions that allow you to organize queries to the database.

The hours of study load H are represented by a set of the following attributes:

,

where is the name of the hour;

– number of hours in units.

Using this entity, you can describe the characteristics for representing the workload in hours. Hours of lecture LH, laboratory LecH and practical PH classes are used in the list of teaching assignments, workload for an individual teacher and IP. Hours of consultations CH, checking individual classes ITH, course projects CPH, credits ZH, exams EH, diploma projects DH, DAK DаcH, scientific research RH, supervision of postgraduate students AH, other types of OH and supervision of practice PrаctH are used in the workload for an individual teacher for a comprehensive description of the teacher's planned work.

The essence of SLR, which represents the rows of the distribution of the training load, is formalized by the following tuple [6]:

,

where – calendar year of the PNN;

–semester of the discipline;

–discipline course;

–number of groups;

–number of students;

–group names;

Dn –name of the discipline;

DP.n –name of the department;

Tn –teacher's name;

Fn –name of the faculty;

T.sd –academic degree of a teacher;

Tp –teaching position;

T.ar –academic title of teacher;

Ts –teacher's rate.

Based on the two presented entities, expressions are formed to extract information from the database with the required information.

The number of hours for lecture delivery is determined as a subquery:

,

where Hq is the number of hours in units;

SlrH.hi – entity identifier field H in the auxiliary entity SLRH.

Number of consultation hours:

.

Number of hours for conducting the training:

.

Number of hours for conducting the program:

.

Number of hours for checking the CTS:

.

Number of hours for CP management:

.

Number of hours for taking credits:

.

Number of hours for taking exams:

.

Number of hours for supervising bachelor's or master's thesis design:

.

Number of hours worked in the DEK:

.

Number of hours allocated to research supervision:

.

Number of hours allocated to supervising graduate students:

.

Number of hours allocated to other forms of educational work:

.

Obtaining the listed information allows you to obtain a set of indicators on the basis of which you can assess the quality of planning the educational load.

Number of hours for teaching lectures and preparing for semester tests:

.

Number of hours for practical classes:

.

The number of hours for course design coincides with the number of classes. The number of hours for supervising diploma design also coincides with the corresponding number.

The number of hours for other hours is formed by finding the sum of all other types of study load:

.

The indicator, which shows the percentage of the number of classes by type to the total number of hours, shows the share of a certain type of workload to the total. For example, to calculate the share of lecture workload, the formula would look like this:

.

It was more difficult to develop mathematical expressions for calculating the distribution indicators by type of occupation. Therefore, the calculation will be presented in the form of a description of the algorithm.

The task of the algorithm is to calculate the number of hours for conducting a ClassType class by teachers of the NumCat category.

Stage 1. Extract records from the database with training load lines corresponding to ClassType.

Step 2. Join the rows with the Teachers table, which contains data about teachers.

Step 3. Filter the data that meets the requirements of the NumCat category.

For category 1, this is either the academic degree "Dr. Tech." or the academic title "Professor."

For category 2, either the academic degree has the value "CTN" or the academic degree has the value "PhD". It is not necessary to check the academic title for this option - a teacher without an academic degree cannot receive the title of "associate professor".

For category 3 – both fields must be empty.

Step 4. Find the sum of the elements of the list obtained in step 3.

End of algorithm.

To calculate the fraction of classes, you need to know the sum of hours. To do this, you need to find the sum of hours of all elements after implementing step 1. Then the fraction is calculated, as in the example above.

The next issue that needs to be resolved is the organization of data storage necessary for calculating the presented indicators.

## 2.3 Organization of information storage for calculating indicators

### 2.3.1 General information about databases and database management systems

A database (DB) is an integrated collection of structured and interrelated data, organized according to specific rules that include general principles for describing, storing, and processing data [15].

A database management system (DBMS) is a set of general or special-purpose software and linguistic tools that provide management of the creation and use of databases [15].

Database development begins with identifying the information needs of users regarding the input and output of the system. The requirements of different user groups are combined in a conceptual data schema [16]. Entities, attributes, and relationships are the constituent elements of the conceptual schema of the data.

An entity is an abstraction of a real-world object, information about which needs to be stored in a database. Entities can correspond to objects (employee, address, document) or processes (train or plane flight, reservation, sale).

An attribute is a characteristic that describes some property of an entity. An attribute that uniquely identifies an entity is called an identifier or key [16].

Based on the entities and the relationships between them, database tables are further formed, after selecting the DBMS format for creating database tables. Columns are created in database tables that actually correspond to the attributes of the entities. At the design stage, it is important to determine the value domains to be filled in, because this helps developers to properly set field constraints and select controls for data entry. When adding records begins, rows are added to the tables. Each row corresponds to a specific object from the subject area.

To create tables, you can use the SQL console, in which you enter queries in the SQL data definition language. This is long. Therefore, either interactive creation tools are used, for example, MySQL Workbench or PHPMyAdmin. These tools provide the ability to create a database in the format of a specific DBMS. But if the DBMS format is only planned before selection, then structural modeling tools such as Erwin Data Modeler or Enterprise Architect or Microsoft Visual Paradigm are used. They help create a logical diagram of the future system, and then generate it in the format of the selected DB.

This was the case before the advent of object-relational mapping or mapping (ORM) technologies [17], which were introduced to eliminate the need to be knowledgeable in a deep understanding of database construction. It is enough for the developer to understand the basics of object-oriented programming, and the corresponding programs converted the class description into tables. Therefore, such an approach, which is used in the work, does not require the construction of database tables.

In the work [6], which is the basis for this study, a database is described that is designed to store training data. But to store statistics, it must be expanded in such a way that it does not affect the operation of other algorithms. The next section defines classes, the use of which will allow storing statistical data.

### 2.3.2 Development of a database structure to support calculations

Based on the type of task being programmed, it can be stated that the results of the distribution indicator calculations should be transferred to the interface for displaying the results. They are not intended for long-term storage, since they lose their relevance the very minute a new version of the RPN is loaded. To perform the task, you can do with local data and calculate the values ​​directly on web pages - tools, for example, pages created based on the Thymeleаf 3 template tag [8], allow you to implement such calculations. But after analyzing the task, we drew attention to the fact that it is advisable to compare the plans with each other in order to be able to choose a more rational one. Therefore, it was decided to develop classes for storing the values ​​of the basic indicators. In addition, creating classes that would allow storing the calculation results is necessary to comply with the Model-View-Control (MVC) concept [18], according to which it is not logical to organize calculations on the View element.

Fig. 2.6. presents the structure of the database, to the tables of which you can add data for calculating statistics. The central class of the structure is the Statistics class, which defines the connections with the RNN variant for which the indicators are calculated. There are two more classes, instances of which will store data on the personal statistics of the teacher's workload distribution StatsValues ​​and group statistics of the department by types of educational activities ClassesStatsValues. The ClassType class is an ENUM type class, the use of which will allow you to conveniently implement data correctness control during processing.

There are quite a lot of fields in the classes, but their composition is justified by the list of tables in which it is proposed to store data calculated based on numbers in the RNN variant.

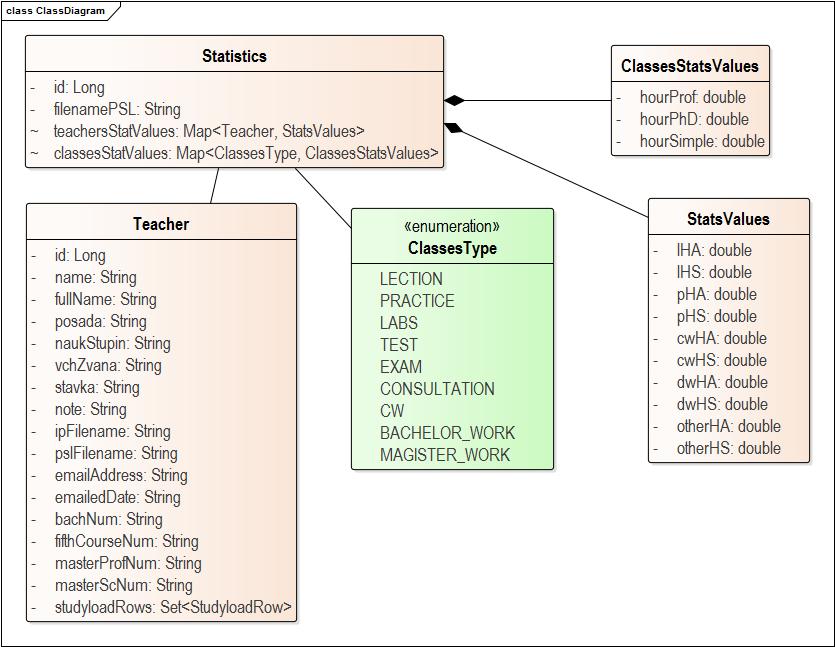


Figure 2.6 – Class diagram for calculating distribution indicators

Next, you need to define the domains for the attributes, for which Enterprise Architect requires you to define the DBMS format. The DBMS format is MySQL8, this is specified in the requirements specification. In any choice, when ORF is used, you need to define the attributes of the mapping of fields and, less often, tables, in order to correctly create the classes of the corresponding project. Below are the descriptions of the classes in tabular form. The “Field Description” column contains Java Persistence API annotations [19], which will allow the Hibernate ORM library [20] to create the DBMS according to the class structure after starting the project.

Based on the "Statistics" class, a "statistics" table will be created, which is intended to store statistical parameters of the next PNN variant. This class contains a set of fields presented in Table 2.1.

Table 2.1 – Description of JPA field markup for the “Statistics” class

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Field name | Field description | Semantics |
| PK | id | @Id@GeneratedValue(strategy = GenerationType.IDENTITY)  private Long | Surrogate table key |
|  | filenamePSL | @Column(nullable = false, unique = true)  String | Path to the file from which the PNN was downloaded |
| FK | Map<Teacher, StatsValues> teachersStatsValues; | @ElementCollection(fetch = FetchType.LAZY)@CollectionTаble(nаme = "teаcher\_stаts", joinСolumns = @JoinСolumn(nаme = "id\_teаcher"))@MаpKeyJoinСolumn(nаme = "id\_stаts") | Description of a class that implements a M:M relationship with teacher workload statistics |
| FK | Map<ClassesType, ClassesStatsValues> classesStatsValues; | @ElementCollection(fetch = FetchType.LАZY)@CollectionTаble(nаme = "clаsses\_stаts", joinСolumns = @JoinСolumn(nаme = "clаsses\_type"))@MаpKeyJoinСolumn(nаme = "id\_stаts") | Description of the table that implements the M:M relationship with workload statistics by class type |

Entity classes for storing statistics are presented in Table 2.2 and Table 2.3. The two specified classes do not have key fields, therefore they reflect a “many-to-many” relationship that implements the presentation of the given sheets with statistics. Table 2.4 presents the interpretation of the values ​​of the fields of the enumerable ClassType class. Based on the entity class “StatsValues”, a database table will be created, designed to store calculations of hours allocated to conducting individual types of classes. Instances of this class will be “connected” to each teacher separately. Based on the entity class “ClassesStatsValues”, a database table will be created, designed to store statistics for the RNN in general.

Table 2.2 – Description of JPA markup for the fields of the “ClassesStatsValues” class

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Field name | Field description | Semantics |
|  | hourProf | @Column(nullable = false) private double | Hours of classes taught by DTN or CTN with the academic title of "professor" |
|  | hourPhD | @Column(nullable = false) private double | Class hours taught by PhD or associate professors |
|  | hourSimple | @Column(nullable = false) private double | Hours of classes taught by teachers only |

Table 2.3 – Description of JPA markup for fields of the “StatsValues” class

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Field name | Field description | Semantics |
|  | lHA | @Column(nullable = false) private double | Lecture Hours (Autumn) |
|  | lHS | @Column(nullable = false) private double | Lecture Hours (Spring) |
|  | pHA | @Column(nullable = false) private double | Hours of work or leisure (fall) |
|  | pHS | @Column(nullable = false) private double | Hours of work or leisure (spring) |
|  | cwHA | @Column(nullable = false) private double | Hours of CP or CR (autumn)ster |
|  | cwHS | @Column(nullable = false) private double | KP or KR hours (spring) |
|  | dwHA | @Column(nullable = false) private double | DP hours (fall) |
|  | dwHS | @Column(nullable = false) private double | DP hours (spring) |
|  | otherHA | @Column(nullable = false) private double | Hours for other types of workload (autumn) |
|  | otherHS | @Column(nullable = false) private double | Hours for other types of loads (spring) |

Table 2.4 – Semantics of enum values ​​of the class “ClassesType”

|  |  |
| --- | --- |
| Field name | Semantics |
| LECTURE | Lectures |
| PRACTICE | Practical classes |
| LABS | Laboratory classes |
| TEST | Credits |
| EXAM | Exams |
| CONSULTATION | Consultations |
| CW | Term papers or projects |
| BACHELOR\_WORK | Bachelor's theses |
| MASTER\_WORK | Master's theses |

All table fields are required. In addition, you can see that the fields are not defined for all indicators of the distribution statistics calculation, but only for the basic ones. All other indicators can be calculated based on the presented ones. This is done because the number of derived indicators may change over time. Based on this, keeping the logic inside the methods is much more profitable in terms of further modification than changing the structure of the database fields.

Also, the StatsValues ​​and ClassesStatsValues ​​tables lack key fields. This is normal for the @ElementCollection table mapping option. This option is used for mapping a many-to-many relationship as an aggregation relationship.

Next, we should describe the choice of a DBMS format for creating a database.

### 2.3.3 Choosing a database management system format

Today, the top 3 database management systems include: Oracle, Microsoft SQL Server, and MySQL. Each of these systems is actively used in various software solutions [21].

Oracle is known as the most powerful DBMS, with excellent scalability and reliability. However, it is rarely used for data storage, because its user license costs more than $ 300. The commercial version of the DBMS is even more expensive. Also, its installation determines high hardware requirements, Oracle is difficult to configure and configure. This option does not suit us because of excessive functionality and high cost of acquisition and maintenance.

The second option is Microsoft SQL Server, which is noted for its ease of use, ease of synchronization with other Microsoft programs (Access, Excel), provides a high level of data protection, and is ready for scaling. But it is also noted for its high license cost: one server is $800). In addition, it is noted that its operation leads to increased resource consumption and limited capabilities for working with web applications, which is critical for us.

The third option is MySQL, which has a number of important advantages for software development: a clear interface, efficient use of resources, the ability to synchronize with other databases, and the availability of a free version.

Given the above, it was decided to develop a database using MySQL. MySQL is a free relational database management system [9]. This open system was created as an alternative to commercial counterparts.

MySQL is one of the most popular database management systems today. MySQL offers a wide range of features that provide a secure environment for storing, maintaining, and retrieving data.

MySQL provides its users with high speed, reliability, and ease of deployment and operation. Many programming languages, such as C++, Perl, PHP, Java, and Python, have programming interfaces for connecting to MySQL from applications written in these languages, which will provide the ability to develop system functionality in different languages.

Thus, this section presents the results of the formalization of the calculation of indicators, resolves the issues of storing the data that will be calculated. This allows us to move on to creating the software.

# 3 Designing a program for calculating workload distribution indicators

## 3.1 Developing a task to create a software solution

### 3.1.1 Product identification

The formulation of the software development task is carried out in accordance with ISO/IEC/IEEE 29148 [22]. According to this document, a Software Requirements Specification (SRS) should be created, which is an analogue of a software development task.

A Software Requirements Specification (SRS) is a detailed description of the requirements for a software system, which is prepared before detailed design begins in order to reduce the number of future changes [22]. This document contains both necessary and sufficient requirements for the successful implementation of the project. To develop requirements, the developer must have a clear and complete understanding of the product, which is achieved by working closely with the project team and the client throughout the development process.

The name of the software being developed is a component for calculating statistics on the distribution of the teaching workload of scientific and pedagogical employees (SPE) of the department “Studyload Stat”, which complements the functional capabilities of the SPE “SLDOCS” teaching workload planning support system (its development and description are presented in [6]) (hereinafter “SLSTAT”).

SLSTAT is intended for use after the preparation of the next version of the RNN NPP. Its function is to calculate the specified distribution parameters in terms of completeness, correctness and quality, in accordance with the specified mathematical support for the calculations and to provide the results in a convenient form for viewing.

### 3.1.2 Glossary

Developing a glossary allows for a common understanding of key terms between the customer (stakeholder) and the software developer.

A department is a structural unit of a faculty that organizes and coordinates teaching and methodological, scientific and educational work in a certain academic discipline or field of knowledge.

A specialty is a specific field of knowledge or professional specialization that a student chooses to study in order to obtain a degree (bachelor's, master's, PhD, etc.).

An academic discipline is a set of knowledge, skills, and abilities that are studied by students within a specific field of science or education.

A group is an association of students who study in the same specialty or program and started their studies at the same time.

An academic hour is the minimum unit of study time, lasting 40-45 minutes. Two academic hours constitute a pair, lecture or practical session.

A teacher is a person who has the appropriate qualifications, knowledge, and experience in their field, and who transmits this knowledge to students through lectures, seminars, laboratory classes, and other forms of instruction.

### 3.1.3 Software solution requirements

Software requirements are divided into functional (what the software should do) and non-functional (how the software or the system to which it is connected should function).  
it is integrated) [23].

Functional requirements define what must be implemented in a product or system, including the actions that users must perform when interacting with the system. Non-functional requirements, on the other hand, specify how the system or software product must function and describe the quality or characteristics that it must have [23].

Functional requirements describe the behavior of a software solution that needs to be developed so that users can effectively perform their tasks according to business needs and individual requirements.

The functionality of this component will be available only to one type of user – the developer of the training load plan. It should be implemented as an additional menu, access to which opens after uploading a new training load option to the database. The existing authentication system provides a sufficient level of access.

During the load calculation, three versions of web pages with statistics should be created:

* page "Summary table of study load" (its template is presented in Fig. 2.1);
* page “Qualitative characteristics of the distribution of teaching workload of teachers” (its template is presented in Fig. 2.2);
* page "Qualitative characteristics of the distribution of teaching workload of teachers by types of classes" (its template is presented in Fig. 2.3).

The presented forms are obtained on the basis of one of the variants of the distribution of the training load, loaded into the database "SLDOCS". This means that after the creation of "SLSTATS" it will be possible to use the data presented in Fig. 2.3, 2.4, 2.5, as a control example for validating the system. In addition, the structure of the tables can serve as a sketch of web pages that should be developed to display the results of calculating statistical parameters.

Typically, UML use case diagrams are used to represent the functionality of the software [24]. But in the work, the task of developing the functionality of this application can be described in the form of a user story. - User story is “a description of the functionality or part of the functionality, written in everyday or business language and showing what the user does or should do. Unlike formal documentation, a User story provides a quick way to process customer requirements without performing administrative tasks related to its maintenance” [25]. User Story was formed as follows: “as a planner for the distribution of the training load, to provide the ability to switch to an exclusively available mode of work with distribution statistics, in which to provide the formation and display of statistics according to the specified layouts with the ability to save to an arbitrary directory on your own disk and send to e-mail”.

As a rule, when talking about non-functional requirements, they most often mean quality attributes. Quality issues are described in the ISO 25010 standard [26]. In this case, non-functional requirements can include:

* productivity;
* convenience of information presentation;
* delimitation of data access.

Performance requirements are not high. It is enough if the system generates a document within 3-5 minutes. This is not a task of operational management, so this time interval will be sufficient. The convenience of presenting information should be manifested in the fact that new pages of the system will have a user-friendly interface (with an easy-to-understand location of buttons, a pleasant design, etc.). The data output on the screen should reproduce the Excel spreadsheets shown above. Delimitation of access to data is necessary, since access to planning quality indicators should be provided to the department management for making organizational decisions. Wide availability is not required for a number of reasons, the discussion of which is beyond the scope of this work. The implementation of this functionality assumes the availability of authentication and authorization tools for the “Training Load Planner” role. The code structure should provide easy modification of the software solution in the future to be able to save statistics in a file of the MS Word document type or (and) an MS Excel spreadsheet with the ability to send by e-mail. In addition, the same technology stack has been defined for this development as for the development of the SLDOCS system.

## 3.2 Defining the technology stack for development

Given that the statistics computing system must be integrated into existing software, there is actually no choice of technology stack – its justification has already been made [6]. The components of the technology stack include:

* Java programming language;
* Spring Framework
* Spring Boot (v.3.0.5);
* Spring Security 6;
* Lombok;
* Hibernate 5.6;
* Java Persistence API (JPA) 3.0;
* Thymeleaf 3;
* BootStrap 4;
* Apache POI;
* Log4j2;
* Maven;
* Junit 5.

The choice of the Java programming language and its Spring framework is natural, because the Spring Framework is an inversion of control container for the Java platform, which has a large set of technologies developed based on it, including the selected Spring Boot [27].

JDK 17 was chosen as the main development language, as it is recognized as one of the newest versions of Java. This is an open source project, so any problems are quickly detected, and the set of features remains relevant. Jakarta EE 9 libraries are used to create the server part [28].

Project Lombok is a Java library that automatically integrates with editor and build tools, improving your Java. Never write another get or equals method again, with a single annotation your class will have a fully functional constructor, automate your logging variables, and much more.

Lombok is a project that adds additional functionality to Java by modifying the source code before Java is compiled. In essence, the Lombok project allows you to get rid of the verbosity of Java in most cases and stop writing large pieces of code with getters, setters, equals, hashcode and toString (although they are usually generated by the IDE, but the programmer still has to read and modify them), as a result Java becomes almost as compact [29].

Implementing the application's data access mechanism usually took a lot of time and effort. To perform even simple queries, implement pagination and auditing, it was necessary to create a lot of standard code. Spring Data JPA is aimed at significantly simplifying and optimizing the implementation of data access levels. To access data in tables associated with some entity class, the developer must create his own repositories (@Repository), which already contain his own methods for manipulating data, and Spring automatically converts them into SQL queries of varying complexity.

The use of the Spring framework is an extension of the idea of ​​ODR – object-relational mapping. The library that implements ODR between relational databases and Java applications is Hibernate [30].

Spring Security is an authentication and access control system that provides a large number of authentication and authorization settings, while freeing the developer from creating autofilters for analyzing the context of web applications. based on Spring It is a recognized security standard for applications, with a focus on authentication and authorization for Java applications [31].

Apache POI is an API for Java applications that allows developers to create, edit, and copy MS Word or MS Excel files using Java programs. It is an open source library developed by the Apache Software Foundation for creating or editing Microsoft Office files in Java. Its inclusion in the project is simply necessary because the report forms are presented as Excel spreadsheet pages. It provides classes and methods for decoding user-entered data or files into formatted MS Office documents [32].

Web pages in the project will be created in the Thymeleаf format [8]. Thymeleаf was created with an orientation to work in conjunction with the Spring framework. It uses html files for its templates. It allows you to apply the following approach to their creation:

1. develop a regular HTML page, having worked out all the issues of data display (in fact, create a sketch);
2. add Thymeleаf validation;
3. add tags that provide data output from @Controller classes passed as attributes or parameters.

Also, page development would be impossible without the use of Bootstrap. The website states that it is a powerful, extensible and multifunctional toolkit for creating a web interface. The ability to use it is becoming an essential skill in the arsenal of a modern web developer [33].

To send the results of the calculation to interested parties, for example, the head of the department, it is necessary to use the Java Mail API - this is a platform- and protocol-independent framework for creating e-mail and messaging programs based on Java technology. It provides abstract classes, support for various protocols and functions, as well as a free help [34].

Log4j2 is a system logging program that outputs specific messages about various situations during operation [35]. For web applications, maintaining such a log on the server is an important functionality for diagnosing system operation and eliminating shortcomings.

Maven is a system for automated project build. Without its use, the developer must independently find and copy to the specified directories the third-party libraries that he needs to implement the required functionality. Maven significantly simplifies the developer's work, as it loads external libraries from a central repository. Maven solves the task of tracking all changes in these libraries, controlling updates. Users define a set of third-party libraries to load by adding the necessary dependency tags to pom.xml, the most current versions of the required libraries will be downloaded and added to the project [36].

For project development, testing and program deployment, the integrated development system Intelij IDEA Ultimate Edition [37] was used. The advantages of this system can be discussed for a long time. Personally, I would like to note the auto-completion functionality, debugging tools, the ability to easily connect to a version control system, and the presence of a large number of additional panels.

An important step in creating software is organizing testing. This is discussed in the next section.

## 3.3 Testing organization

Test-driven development (TDD), also known as a test-driven approach to software development, is a development methodology based on a cyclic process. The first step is to write a test that reflects the expected result. The next step is to create code that implements the desired functionality and passes the previously written test. This is followed by optimizing the generated code by constantly passing all the tests, which is called refactoring. The cyclic development process continues until the desired result is achieved [38].

Software testing is a process whose task is to verify the correctness of the created code. Testing involves the input of previously prepared data, while the results that the developed code should verify are known. If the results correspond to the expected result, the test is considered to have been successfully passed.

This Test-driven development (TDD) methodology involves creating pipelines for automatic testing of the created software product code by creating unit, system, and functional tests that define the requirements for the code before it is written. That is, tests are created before the code to create an idea of ​​how the program should function. When direct development begins, immediately after completion, the developer simply uploads a new version to the project repository. After successfully passing the test, the written code is optimized if necessary. In this case, each change occurs through the passage of tests, which is a simpler and more reliable approach. This approach provides constant quality control of the code and allows you to confidently optimize it.

A unit test is used to test a single, logically separated and isolated unit of the system, most often a class method or a simple function. The isolation of the unit under test is ensured by using stubs.

A comprehensive test allows you to check the interaction of several system components, perceiving the system as a black box. In this case, a module can be considered as a collection of components united by a facade that has a suitable API for testing. Communication isolation is achieved using stubs.

The TDD methodology includes the following steps:

1. Writing a test for new functionality or reproducing an existing bug.
2. Running all tests to make sure the new test fails.
3. Creating code that will ensure that the test passes.
4. Running tests to ensure that they all pass successfully. Passing a new test confirms the implementation of a new feature or bug fix, and passing other tests confirms the correct operation of previously implemented functionality.
5. Refactoring and optimizing code to improve its supportability and performance.
6. Re-running the tests to make sure they still pass.
7. Return to point 1.

The use of TDD allows you to look at the creation of program class and method code differently, keeping in mind the testing processes, adapting the organization of classes and methods to this. The ideology of task statement also changes, so when using TDD, the task statement turns into the language of automatic tests, using which compliance with requirements will be checked. At the same time, if the tested code passes TDD tests, then it is considered ready, without analyzing how it is written. If the code passes the tests, but the component does not work as expected, this is a reason to add new tests.

Below is an example of a test that verifies the correctness of the written method responsible for creating workload for teachers. This test checks whether the established styles are followed and whether the document is correctly formatted using the Apache POI framework.

The architecture of software products developed using TDD assumes that the functionality is thoughtfully distributed in the form of separate methods, and into minimally simple procedures. Therefore, the stability of programs developed using TDD is higher, because all the functionality of the program is covered by tests.

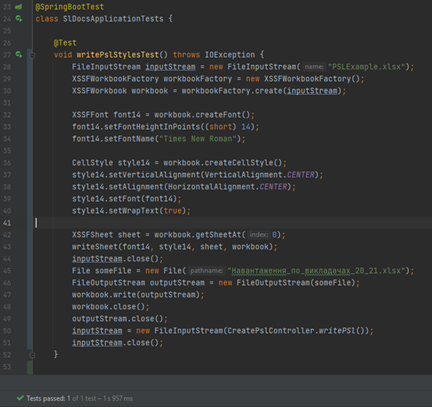


Figure 3.1 – Unit test written using SpringBoorTest

The maintainability of projects in which everything or almost everything is tested is very high, as developers can not worry about making changes to the code, because the results of automated testing will detect any errors.

The next section discusses options for building the system and describes how to use it.

# 4 Description of preparation for application and use of the software solution

## 4.1 Description of preparing the software solution for use

The developed software solution is a new version of the "SLDOCS" software, to which a statistics calculation block has been added. Based on this, it is necessary to consider typical approaches for deployment for web applications created on the basis of Jakarta EE, working with a database in the MySQL DBMS format.

Web application

Software deployment can be done according to two possible options:

1. individual use;
2. shared use.

Individual use involves deploying all components of the web application operating environment (web server, DBMS, browser) on one computer. For shared use, only the browser is installed on the user's computer, and the other parts are deployed on remote equipment. Also, in shared use, it is necessary to implement separate roles in the system to provide access to the formation of statistics. This is determined by the fact that statistical information is needed primarily by the department management, as well as heads of specialties and specialty guarantors.

The deployment diagram [39] for an individual use case of the developed software is presented in Fig. 4.1.

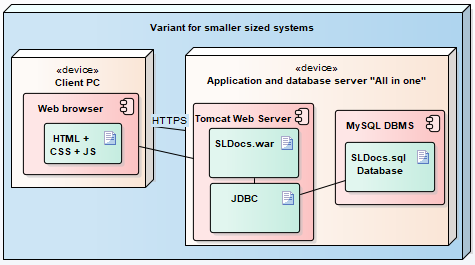


Figure 4.1 – Software deployment diagram for individual use

The deployment diagram for the shared use case of the developed software is presented in Fig. 4.2.

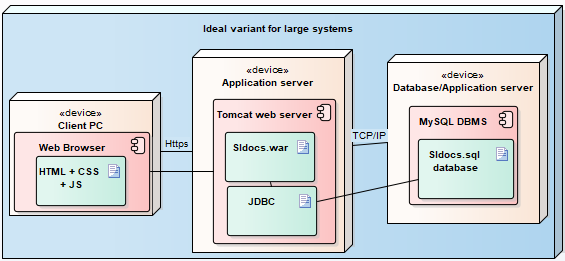


Figure 4.2 – Shared Software Deployment Diagram

In the conditions of combat operations, the advantages of cloud-based services were felt. Deploying an application in a cloud service allows you to automate the “delivery” processes, which allows you to think only about uploading a new version of the software to the GitHub cloud repository. The rest of the work is done by the cloud services themselves. With this approach, you can create new versions of the system even every day, because for a nominal fee the service takes on all the work [40]. Since data processing occurs on the server, you can easily add additional servers to process an increased amount of data, if necessary. In addition, thanks to the centralized data storage, data management and ensuring its integrity becomes much easier.

In the case of individual use, you will have to download a new version of the web application either yourself, following the instructions, or using tools that allow you to control the computer remotely (for example, Teamviewer [41].

To deploy the server part of the system, the Apache Tomcat 10.1.12 servlet container was used [42]. This version already works on Jakarta EE 9 language components. On the one hand, this is a step forward, and on the other hand, it requires reviewing the code of other parts of the web application, because they will not work on this server. Apache Tomcat was chosen because this web server is the most common among cloud services, along with the MySQL database. That is, its choice is important for joint use. For an individual, a file with statistics is created, which can either be opened on disk or sent to the email address that the user enters in the dialog. Another advantage of the individual option is that its operation requires fewer resources for deployment and support, since there is no need for additional servers to process large amounts of data. This option is much cheaper in terms of infrastructure costs. It is also easier to organize user authorization in it.

Minimum client computer requirements:

* Intel Core i5 or i5, Intel Xeon or Intel Core Duo 2.3 GHz processor;
* any type of operating system;
* Internet access;
* having a browser that supports HTML5.

As for hardware requirements, again this depends on the deployment option. In shared use, this issue does not need to be addressed at all - ensuring the installation capability will be determined by the cloud service owners themselves.

In the case of individual deployment, the computer requirements are determined by the requirements for the most "demanding" element among the three - the browser, Apache Tomcat 10.1, the MySQL 8 database server. The winner is the MySQL 8 database server, the requirements for deployment of which are the highest. The hardware requirements for installing the MySQL server are as follows:

* RAM: 3 GB (8 GB offered);
* graphics accelerators: nVidia or ATI with support for OpenGL 1.5 or higher;
* Display resolution: 1280 × 1024 recommended, minimum 1024 × 768.

To access the program, you need to open a browser and go to the address:https://{hostname}:{port}/SLDOCS/.

It is worth noting that any modern laptop or desktop computer is suitable for deployment, which can be selected for the example here [43].

## 4.2 Description of the procedure for using the software functionality to generate statistics

To start working with the system, it is necessary to go through the authentication process, because unauthenticated users cannot work in the system. This is implemented through the use of Spring Security. This is a necessary step that ensures the security and personalization of the service. After successful authentication, the user gains access to all the functions of the application. The interface for entering the user's login and password into the system is presented in Fig. 4.3.

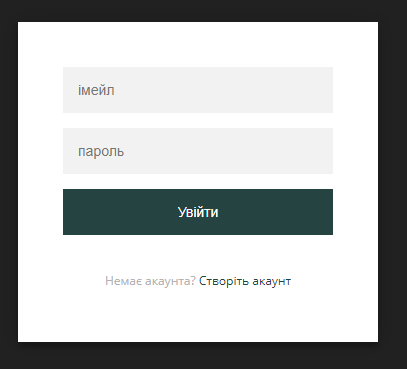


Figure 4.3 – Login page

After successful authentication, the user is automatically redirected to the main page, where brief instructions on how to create a  
extracts (see Fig. 4.4).

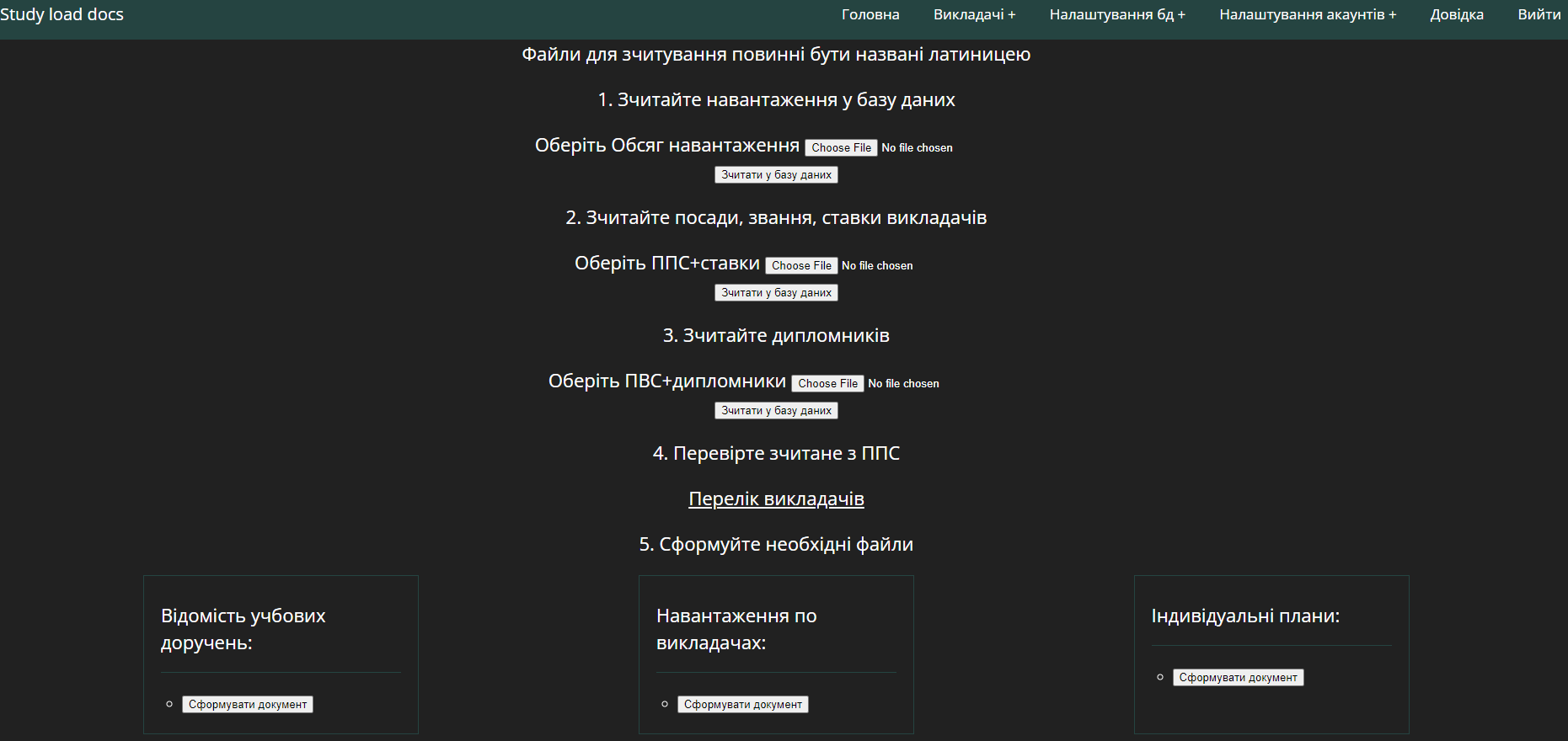


Figure 4.4 – Home page of the “SLDOCS” application

The process of downloading the training load plan is discussed in [6]. Creating individual plans and personal extracts takes from 15 to 45 seconds. Until this process is completed, the buttons for downloading documents will not be visible on the page. They will be displayed after the creation is complete.

The button for calculating statistics will appear only after downloading the required document. The download consists in reading the Excel file with the distribution of the training load, after which the read data will be written to the corresponding database tables. The central page will look like shown in Fig. 4.5.

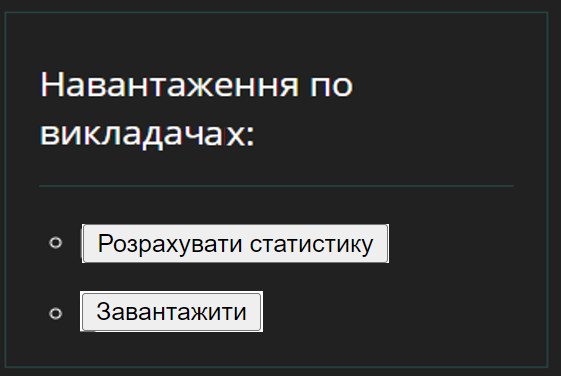


Figure 4.5 – Interface for generating files

If you click on the "Calculate Statistics" button, you will be taken to a page that looks like the one in Fig. 4.6. A simple interface will allow you to enter the required data, after which you will receive three pages with the load indicators that were determined by the technical specifications. The calculation results can be seen in Fig. 4.7, Fig. 4.8 and Fig. 4.9.

It can be seen that the resulting pages correspond to the presented data, the values ​​on the pages presented in Fig. 2.3, Fig. 2.4, Fig. 2.5. This allows us to state that the calculation algorithms are implemented in the correct way.

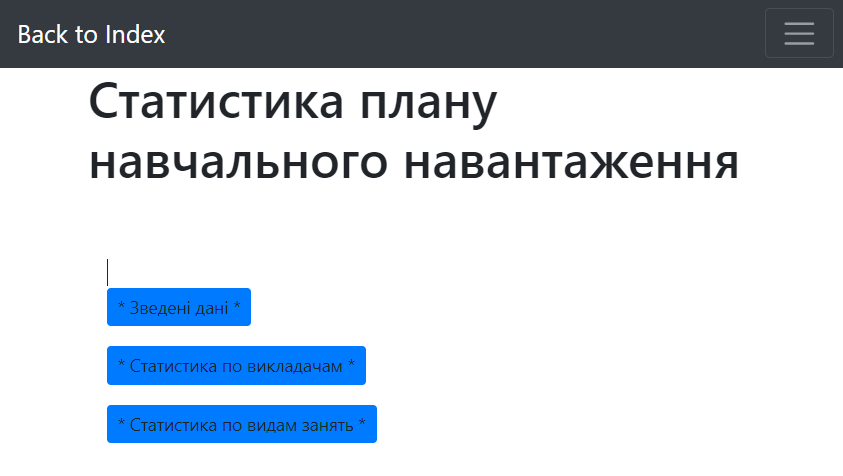


Figure 4.6 – Page for starting calculations

On the page with general data, you can see the elements of analytical analysis of the obtained figures. In the case when the teacher “does not have enough” of the teaching load, these values ​​are highlighted in red. This is achieved by using the th:if tags provided by Thymeleаf. Using such tags, as well as the tags of cycles and multiple-choice, you can implement the desired “coloring” of the page, which will allow you to better analyze statistics.

On each page with calculated indicators, there is a "Send" button. At the beginning of the work, the system offers to enter a valid email address, to which the user then receives an email with the presented summary data.  
in Excel format.

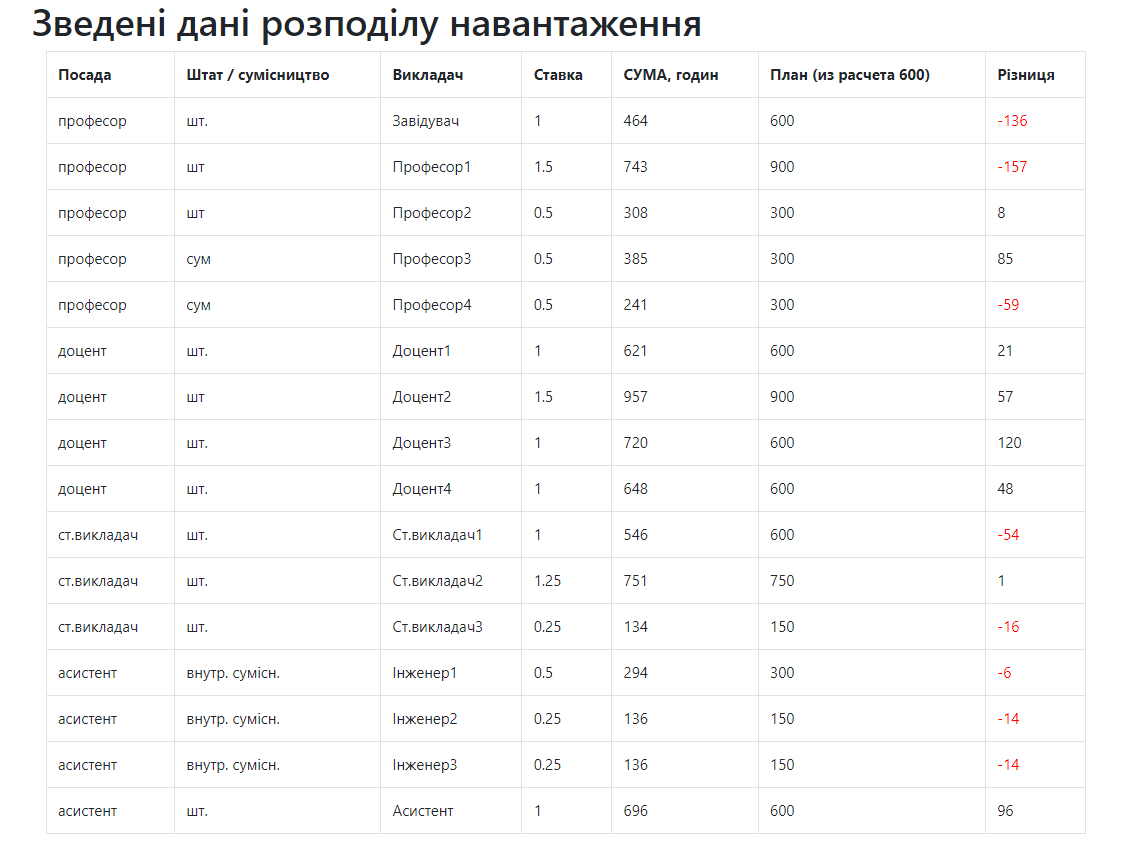


Figure 4.7 – Summary page of study load distribution

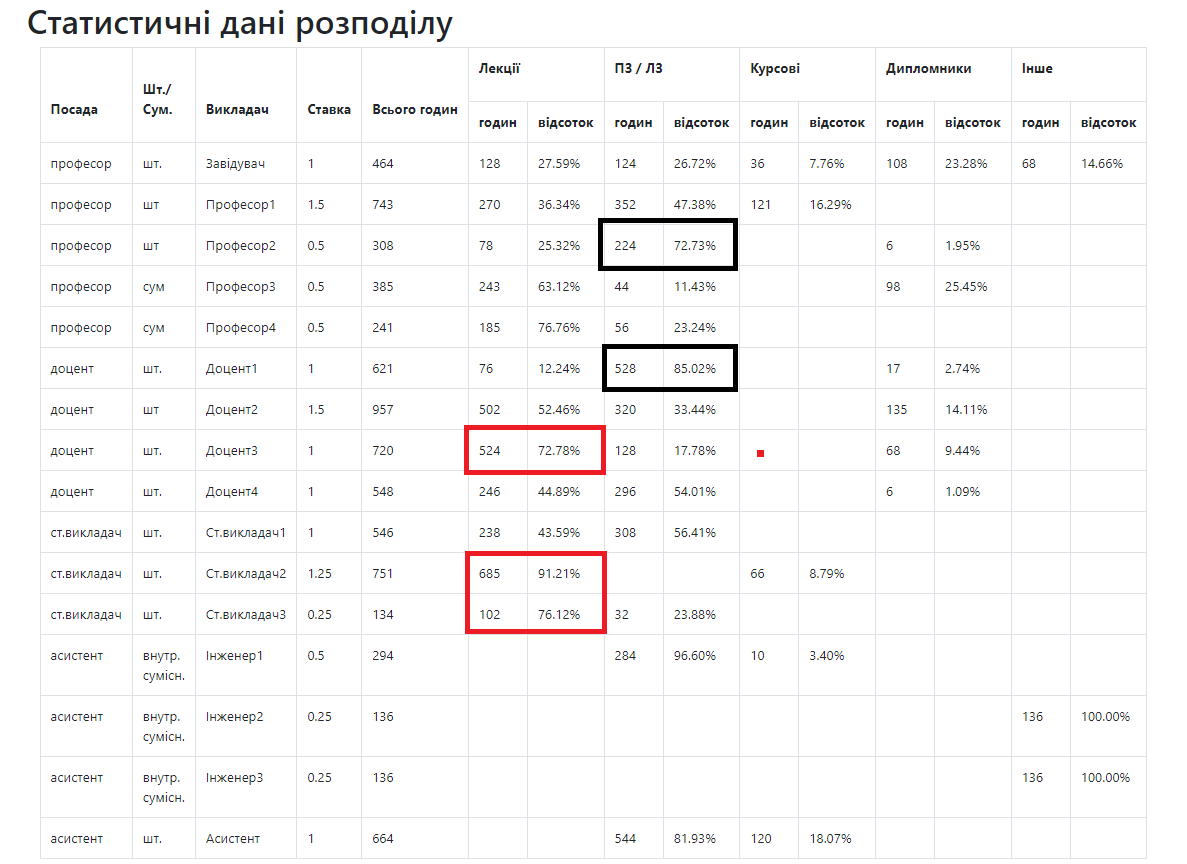


Figure 4.8 – Information on the quality of the distribution of the teaching load



Figure 4.9 – Load distribution information

# 5 Economic justification of the software development project

## 5.1 Justification of the feasibility of software development

This section presents the economic justification for software development foranalysis of the quality of the distribution of the teaching load of the teachers of the department of NTU "KhPI", which will be carried out in accordance with [33].The use of software provides users with the opportunity to form and evaluate the quality of the formation of the teaching load of teachers. Due to the large number of academic disciplines taught by the teachers of the department, there is a need to develop programs to automate routine work on the formation of teaching load plans. An integral part is the assessment of the quality of such a load and the formation of recommendations for its improvement.

The users of the developed software will be employees of the department of PIITU NTU "KhPI".

The development of software that will allow you to easily and quickly form and analyze the teaching load will help increase the efficiency of managing the educational process, prevent errors related to overloading or underloading teachers, and reduce the time required to correct them.

## 5.2 Assessment of software competitiveness in comparison with its analogue

To analyze and assess the competitiveness of the developed software, the university's ACS, which was developed by the Research and Development Department of Applied Information Technologies [[34](https://vuz.osvita.net/)]

The competitiveness of software is determined by its properties as a product on the market. To assess the competitiveness of the developed software, an analysis and comparison with the selected analogue is carried out in terms of functional purpose, main technical and operational parameters, and areas of use.

The operational and technical level (OTL) of the developed software is determined by the operational and technical level index (OTL) using the point-index method on a five-point evaluation scale. The results of calculating OTL are given in Table 5.1.

Table 5.1 – Calculation of the quality indicator

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Software product quality indicators | Weighting factor, | Project | | Analog | |
|  |  |  |  |
| Ease of use (user interface) | 0.1 | 3 | 0.3 | 4 | 0.4 |
| Novelty (compliance with modern requirements) | 0.1 | 5 | 0.5 | 4 | 0.4 |
| Compliance with the customer's activity profile | 0.25 | 5 | 1.25 | 3 | 0.75 |
| Resource efficiency | 0.05 | 4 | 0.20 | 3 | 0.15 |
| Reliability (data protection) | 0.2 | 3 | 0.6 | 2 | 0.4 |
| Data access speed | 0.1 | 4 | 0.4 | 3 | 0.3 |
| Flexibility in customization | 0.15 | 5 | 0.75 | 2 | 0.3 |
| Ability to train staff | 0.05 | 3 | 0.15 | 1 | 0.05 |
| Overall quality score | | 3.25 | | 2.75 | |

For the software being compared and the software product as its analogue, the coefficient of the technical level will be equal to 𝐴𝑘 = 3.25 ⁄ 2.75 = 1.18. Since the coefficient is greater than 1, the development of the software is justified from a technical point of view.

## 5.3 Justification of the feasibility of software development

The project manager and the executor (programmer) participate in the software development. The scope of work for the software development project is given in Table 5.2. The schedule of work is given in Table 5.3.

Table 5.2 – Software development project scope of work

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Contents of the works | Performer | Duration, day | Load | | | |
| Duration, day | % | People no-day | |
| 1. Justification of the need to create an application | | | | | | |
| 1.1. Description of the subject area | | | | | | |
| 1.1.1 Basic concepts | Manager, Programmer | 2 | 1  2 | 10  100 | 0.1  2 | |
| 1.1.2 Software analysis | Programmer | 5 | 5 | 100 | 5 | |
| 1.1.3 Description of software development stages | Manager, Programmer | 3 | 1  3 | 25  100 | 0.25  3 | |
| 1.1.4 Choosing a development methodology | Manager, Programmer | 2 | 1  2 | 20  100 | | 0.2  2 |
| 1.2. Description of application design | | | | | | |
| 1.2.1 Business requirements analysis | Programmer | 2 | 2 | 100 | | 2 |
| 1.2.2 Analysis, improvement of business processes | Manager, Programmer | 4 | 2  4 | 25  100 | | 0.5  4 |
| 1.2.3 Functional requirements | Manager, Programmer | 2 | 1  2 | 50  100 | | 0.25  2 |
| 1.2.4 Non-functional requirements | Manager, Programmer | 2 | 1  2 | 50  100 | | 0.25  2 |
| 1.3 Problem statement | Head,  Programmer | 5 | 2  5 | 20  100 | | 0.4  5 |
| Together with stage 1 | Manager, Programmer | 27 | 9  27 | 33  100 | | 2.45  27 |
| 2 Overview of methods and approaches for solving the problem | | | | | | |
| 2.1 Overview of existing software | Manager, Programmer | 3 | 1  3 | 25  100 | | 0.25  3 |
| 2.2 Software architecture | Manager, Programmer | 6 | 2  6 | 25  100 | | 0.5  6 |
| Together with stage 2 | Manager, Programmer | 9 | 3  9 | 33  100 | | 0.75  9 |
| 3 Software design | | | | | | |
| 3.1 Development of diagrams | Programmer | 4 | 4 | 100 | | 4 |
| 3.2 Database development | Programmer | 10 | 10 | 100 | | 10 |
| 3.3 Choosing a technology stack | Programmer | 5 | 5 | 100 | | 5 |
| Together with stage 3 | Programmer | 19 | 19 | 100 | | 19 |

End of table 5.2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Contents of the works | Performer | Duration, day | Load | | | |
| Duration, day | % | People no-day | |
| 4 Software development and testing | | | | | | |
| 4.1 Features of software implementation | Programmer | 20 | 20 | 100 | | 20 |
| 4.2 Software testing | Programmer | 3 | 3 | 100 | | 3 |
| 4.3 Software evaluation | Manager, Programmer | 2 | 1  2 | 50  100 | | 0.5  2 |
| 4.4 Software Improvements | Manager, Programmer | 2 | 1  2 | 50  100 | | 0.5  2 |
| Together with stage 4 | Manager, Programmer | 27 | 2  27 | 7  100 | | 1  27 |
| 5 Preparation of working documentation | | | | | | |
| 5.1 Economic justification of the project | Programmer | 3 | 3 | 100 | | 3 |
| 5.2 Preparation of an explanatory note | Manager, Programmer | 15 | 6  15 | 20  100 | | 1.2  15 |
| Together with stage 5 | Manager, Programmer | 18 | 6  18 | 33  100 | | 1.2  15 |
| Together for the project | Manager, Programmer | 100 | 14  100 | 14  100 | | 5.4  100 |

Table 5.3 – Project work schedule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Content of the work | Performer | Duration, day | Work schedule | |
| Beginning | End |
| 1. Basic concepts | Head | 1 | 02/15/2024 | 02/15/2024 |
| Programmer | 2 | 02/15/2024 | 02/16/2024 |
| 2. Analysis of the patient card management system | Programmer | 5 | 02/17/2024 | 02/21/2024 |
| 3. Description of development stages | Head | 1 | 02/22/2024 | 02/22/2024 |
| Programmer | 3 | 02/22/2024 | 02/24/2024 |
| 4. Choosing a development methodology | Head | 1 | 02/25/2024 | 02/25/2024 |
| Programmer | 2 | 02/25/2024 | 02/26/2024 |
| 5. Business requirements analysis | Programmer | 2 | 02/27/2024 | 02/28/2024 |
| 6. Analysis and improvement of business processes | Head | 2 | 02/29/2024 | 01.03.2024 |
| Programmer | 4 | 02/29/2024 | 03.03.2024 |
| 7. Functional requirements | Head | 1 | 04.03.2024 | 04.03.2024 |
| Programmer | 2 | 04.03.2024 | 05.03.2024 |
| 8. Non-functional requirements | Head | 1 | 06.03.2024 | 06.03.2024 |
| Programmer | 2 | 06.03.2024 | 07.03.2024 |
| 9. Problem statement | Head | 2 | 08.03.2024 | 09.03.2024 |
| Programmer | 4 | 08.03.2024 | 03/12/2024 |

End of table 5.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Content of the work | Performer | Duration, day | Work schedule | |
| Beginning | End |
| 10. Review of existing systems | Head | 1 | 03/13/2024 | 03/13/2024 |
| Programmer | 3 | 03/13/2024 | 03/15/2024 |
| 11. Software solution architecture | Head | 2 | 03/16/2024 | 03/17/2024 |
| Programmer | 6 | 03/16/2024 | 03/21/2024 |
| 12. Developing sequence diagrams | Programmer | 4 | 03/22/2024 | 03/25/2024 |
| 13. Database development | Programmer | 10 | 03/26/2024 | 04.04.2024 |
| 14. Choosing a technology stack | Programmer | 5 | 04/05/2024 | 09.04.2024 |
| 15. Features of software implementation | Programmer | 20 | 10.04.2024 | 04/29/2024 |
| 16. Software Testing | Programmer | 3 | 04/30/2024 | 02.05.2024 |
| 17. Software Evaluation | Head | 1 | 03.05.2024 | 03.05.2024 |
| Programmer | 2 | 03.05.2024 | 04.05.2024 |
| 18. Software Improvement | Head | 1 | 05.05.2024 | 05.05.2024 |
| Programmer | 2 | 05.05.2024 | 06.05.2024 |
| 19. Economic justification of the project | Programmer | 3 | 07.05.2024 | 09.05.2024 |
| 20. Preparation of an explanatory note | Head | 6 | 10.05.2024 | 05/15/2024 |
| Programmer | 15 | 10.05.2024 | 05/24/2024 |

The project schedule in the form of a Gantt chart is presented in Fig. 5.1.

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Figure 5.1 – Project timeline

## 5.4 Calculating project costs for software development

The basic salary of the project performers is indicated in Table 5.4. The time spent on software development for each performer is taken based on his workload according to the work schedule.

Table 5.4 – Basic salary of project executors

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Position | Position salary, UAH | Average daily rate, UAH | Development time costs, man-days | Basic salary expenses, UAH |
| Head | 8500 | 8500/21 = 405 | 5.4 | 2187 |
| Programmer | 1950 | 1950/21 = 93 | 100 | 9300 |
| Total | | | | 11487 |

Calculations of the use of machine time required for software development were made. The cost of one hour of machine time is 5 UAH/hour, and the multiprogramming coefficient is taken as 1.

;

.

The costs of materials are given in Table 5.5.

Table 5.5 – Material costs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Materials | Unit of measurement | Required quantity | Price per unit, UAH | Amount, UAH |
| General notebook | Pcs. | 1 | 50 | 50 |
| Office paper | Bundle | 1 | 100 | 100 |
| Printer toner | Pcs. | 1 | 200 | 200 |
| Total | | | | 350 |

Therefore, the capital investment of the project is equal to:

.

The estimate for software development is given in Table 5.6.

Table 5.6 – Estimate for software development

|  |  |
| --- | --- |
| Expense items | Amount, UAH |
| Basic salary | 11487 |
| Additional salary | 5744 |
| Deductions | 3790 |
| Material costs | 350 |
| Machine time costs | 4000 |
| Organization overhead costs | 6892 |
| Total | 32263 |

So, the main costs for software development amounted to 32,263 UAH.

## 5.5 Calculating the cost of software implementation

Below are calculations of the costs of software implementation.

Necessary funds for the purchase of basic equipment, which includes a PC worth 30,000 UAH. The effective annual fund of operating time of the technical means was taken as 2008 hours, that is, 251 working days of 8 hours of work. The laboriousness of one-time information processing is taken as 6 hours of machine time. The frequency of the solution is 251 days per year.

The costs of implementing the project are:

.

Therefore, the total costs of the software development project will be:

.

## 5.6 Calculating the cost of software implementation

Let's calculate the total costs associated with implementing the analogue with which the software is compared. Such costs consist of:

* costs of purchasing a software product$80 per month for the "unlimited" version, or approximately UAH 38,400 per year;
* equipment costs, you will need the same PC as for software worth UAH 30,000.

Capital investments consist of:

.

Therefore, the total costs of implementing the analogue will be:

.

You also need to take into account that 38,400 UAH will need to be paid each year, so for example, the amount for the second year of implementation will be 99,300 UAH, and for the third year 137,700 UAH.

## 5.7 Calculation of current operating costs of software and its analogue

Operating costs include costs associated with ensuring the normal functioning of the software.

Data on the salaries of software users are presented in Table 5.7. Data on the salaries of specialists for the implementation of the analogue are presented in Table 5.8.

Table 5.7 – Salary data of “Client card” users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Position | Position salary, UAH | Average daily rate, UAH/day | Time spent working with software, man-days | Salary fund, UAH |
| Programmer | 22000 | 1047.62 | 1 day \* 12 = 12 | 12571 |
| Administrator | 12000 | 571.43 | 8 days \* 12 = 96 | 54857 |
| Together | | | | 67428 |

Table 5.8 – Data on salaries of specialists for the implementation of the analogue

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Position | Position salary, UAH | Average daily rate, UAH/day | Time spent working with software, man-days | Salary fund, UAH |
| Programmer | 40000 | 1904.76 | 24 | 45714 |
| Administrator | 18000 | 857.14 | 96 | 82286 |
| Together | | | | 128000 |

The costs of salaries for software users and similar are:

;

.

The depreciation amounts for software and similar are:

.

Currently, the electricity tariff is 2.64 UAH/kWh. The installed capacity for the computer is 0.4 kW.

Electricity costs for users of the software and analogue are:

.;

.

The costs for current repairs of equipment, software and similar will be:

*.;*

.

The cost of materials consumed each year is 1 percent of the book value of the main equipment. Therefore, the cost of materials for the software and HospApp users is the same, namely:

The overhead rate is 20% of direct costs. Overhead costs for software and similar will be:

.

.

The annual operating costs of the software are presented in Table 5.9.

Therefore, the current operating costs of the software will be 143,527 UAH, and its analogue - 268,226 UAH.

Table 5.9 – Annual operating costs of a software product

|  |  |  |
| --- | --- | --- |
| Expense items | Software costs, UAH | HospApp costs, UAH |
| Basic and additional salary with deductions | 115168 | 218624 |
| Amortized deductions | 2581 | 2868 |
| Electricity costs | 912 | 1013 |
| Current repair costs | 645 | 717 |
| Material costs | 300 | 300 |
| Overhead | 23921 | 44704 |
| Total expenses | 143527 | 268226 |

## 5.8 Calculation of the economic effect of software development

The annual economic effect of using the developed software is determined by the difference between the reduced costs of developing and operating the software per year and the costs of purchasing and operating its analogue.

The normative coefficient of economic efficiency is 0.33

The given costs for performing a unit of work using the developed software and analogue are:

.

.

Then the economic effect of using the developed software will be:

.

Summary calculations of the economic effect are presented in Table 5.10.

Table 5.10 – Annual operating costs of a software product

|  |  |  |
| --- | --- | --- |
| Characteristic | Value for | |
| Software | analogue |
| Cost of work (current operating costs of the software product), UAH. | 143527 | 268226 |
| Total costs associated with software implementation, UAH. | 54763 | 60900 |
| The costs per unit of work are given, UAH. | 161598 | 288323 |
| Economic effect from using the software product, UAH. | 299719 | |

After determining the annual economic effect, it is necessary to calculate the payback period for the costs of developing the software product.

The payback period for the software will be:

.

The calculated coefficient of economic efficiency of the development is:

.

The actual coefficient of economic efficiency of software development is greater than the normative one (5.56>0.33), therefore, software development and implementation is effectively feasible.

Thus, the technical feasibility and economic efficiency of the developed software have been proven.

# Conclusions

As a result of the thesis, a new algorithm for distributing the educational load was proposed, in which a stage of calculating indicators appeared that can be analyzed in order to assess the completeness, correctness and quality of the resulting distribution option. To automate the business process, the SLDOCS web application was added with the SLSTATS component, which provides calculation and display of certain statistics. When creating a new software component, the following tasks were performed:

1. based on communication with workload planners and personal participation in the preparation of the workload plan, a BPMN diagram of the improved business process of creating a workload plan with assessment was created;
2. requirements for the software component are defined;
3. software design, implementation and testing completed;
4. Studies were conducted on the application of the created software solution for assessing the distribution of the educational load.

Based on the results of the work, the following theses are proposed for discussion.

1. Automatic determination of certain indicators of the study load plan when planning the department's activities helps ensure compliance with accreditation and licensing requirements. This is due to the fact that managers will be able to quickly receive indicators of the quality of the distribution and timely intervene in the planning processes, ensuring the appropriate indicators.
2. Using the Spring framework is effective for creating information systems based on a multi-tier architecture.
3. Using the MySQL 8 database format for data storage makes it possible to store all data for free.
4. The developed software component is appropriate for use in the process of distributing the department's teaching workload.

The skills acquired during the period of education were used to create the web application, which indicates the excellent construction of training plans at the department. The process of creating programs also showed the importance of high-quality development of both the requirements and the software project.

The totality of the presented results and conclusions allows us to state that all the tasks set for the thesis project were successfully completed, and the goal of the work should be considered largely achieved.

# references

1. Law of Ukraine "On Higher Education" // https://zakon2.rada.gov.ua-/laws/show/1556-18, 04/23/2024.
2. Order of the Ministry of Education and Science of June 21, 2022 No. 686 "On approval of time standards for planning and accounting of educational work and lists of types of educational, methodological, innovative, scientific, organizational work and other pedagogical activities of pedagogical and scientific and pedagogical workers of institutions of professional pre-higher education."
3. Regulations on planning and accounting for the main types of work of scientific and pedagogical employees of NTU "KhPI"; approved by the Academic Council of NTU "KhPI" on September 23, 2022, protocol No. 6; put into effect by order of the rector on September 28, 2022 No. 264 OD. - Kharkiv, 2022. - 28 p.
4. Repiv V. Process approach to business process modeling – Kyiv: Dumka, 2019. – 522 p.
5. Dvukhglavov, D., Bezmenov, M., Shmatko, O., Dvukhglavova, A., Korkhovy, D., & Cherkun, M. (2022). Formalization of the process of preparing the teaching load plan of the teachers of the department of the higher education institution / Bulletin of the National Technical University "KhPI". Series: System Analysis, Management and Information Technologies (1 (7)), 35–45. //<https://doi.org/10.20998/2079-0023.2022.01.06>, 10.04.2024.
6. Korkhovy D.S. Design, development and research of a software solution for forming extracts from the study load plan of scientific and pedagogical employees of the department: master's thesis. – Kh.: NTU "KhPI", 2023. – 100 p.
7. Official site of the Spring // https://spring.io/, 19.05.2024.
8. Official website of the[Thymeleaf](https://www.thymeleaf.org/)//<https://www.thymeleaf.org/>, 05/19/2024.
9. [BPMN Specification - Business Process Model and Notation](https://www.bpmn.org/)// https://www.bpmn.org/, 31.03.2024.
10. [UML modeling tools for Business, Software, Systems and Architecture (sparxsystems.com)](https://sparxsystems.com/)//<https://sparxsystems.com/>, 31.03.2024.
11. Embley D, Thalheim B. Handbook of Conceptual Modeling: Theory, Practice, and Research Challenges. Berlin, Springer, 2017. – 597 p.
12. What is an ORM? How Does It Work? How Should We Use One? | Baeldung on Computer Science //<https://www.baeldung.com/cs/object-relational-mapping>, 04/27/2024.
13. MVC - MDN Web Docs Glossary: ​​Definitions of Web-related terms | MDN (mozilla.org) //<https://developer.mozilla.org/en-US/docs/Glossary/MVC>, 04/27/2024.
14. Guide to JPA with Hibernate - Relationship Mapping (stackabuse.com) //https://stackabuse.com/a-guide-to-jpa-with-hibernate-relationship-mapping/ , 04/27/2024.
15. [Hibernate Tutorial (javaguides.net)](https://www.javaguides.net/p/hibernate-tutorial.html) <https://www.javaguides.net/p/hibernate-tutorial.html>, 05/31/2024.
16. [Database comparison //](http://ni.biz.ua/8/8_6/8_63622_sravnenie-subd.html) <http://ni.biz.ua/8/8_6/8_63622_sravnenie-subd.html>, 08.03.2024.
17. ISO/IEC/IEEE 29148. Systems and software engineering – Life cycle processes – Requirements engineering //<https://webstore.iec.ch/preview/>info\_isoiecieee29148%7Bed2.0%7Den.pdf, 03/23/2024.
18. Requirements engineering for software and systems / Phillip A. Laplante. - CRC Press, 2018. - 399 p.
19. Fowler M. UML Distilled. Ch. 4 //[https://www.pearsonhighered.com/assets/samplechapter/0/3/2/1/0321193687.pdf](https://www.pearsonhighered.com/%20assets/samplechapter/0/3/2/1/0321193687.pdf), 06.02.2024.
20. ISO/IEC 25010:2011(en). Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models.
21. Spring Boot/Cloud and WildFly Swarm comparison //<https://blog.akquinet.de/2017/01/12/the-lightweightness-of-microservices-comparing-spring-boot-wildly-swarm-and-haskell-snap/#:~:text=Spring%20Boot%20provides%20a%20single,Java%20are%20built%20with%20Maven>, 04/23/2024.
22. Site about Jakarta EE 9 //<https://jakarta.ee/release/9/>, 04/23/2024.
23. [Project Lombok](https://projectlombok.org/) <https://projectlombok.org/>, 03/23/24.
24. [Spring Security](https://spring.io/projects/spring-security)//<https://spring.io/projects/spring-security>, 03/23/24.
25. Apache POI // https://poi.apache.org, 02.02.2024.
26. [Bootstrap · The most popular HTML, CSS, and JS library in the world. (getbootstrap.com)](https://getbootstrap.com/)//https://getbootstrap.com/, 03/23/24.
27. [Log4j – Apache Log4j™ 2](https://logging.apache.org/log4j/2.x/)//<https://logging.apache.org/log4j/2.x/>, 03/23/2024.
28. [Maven - Welcome to Apache Maven](https://maven.apache.org/)//<https://maven.apache.org/>, 03/23/2024.
29. IntelliJ IDEA – the Leading Java and Kotlin IDE // https://www.jetbrains.com/idea/, 10.05.2024.
30. Beck K. Test-driven development: By example. Ch. 25 //<https://dokumen.pub/test-driven-development-by-example-0321146530-9780321146533-n-2465731.html>, 06.05.2024.
31. The Perceived Business Benefit of Cloud Computing: An Exploratory Study //[https://scholarworks.lib.csusb.edu/cgi/viewcontent.cgi?referer=&httpsredir=1& article=1297& context=jitim](https://scholarworks.lib.csusb.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&%20article=1297&%20context=jitim)b, 04/23/2024.
32. Apache Tomcat 9 // https://tomcat.apache.org/, 02.12.2024.
33. Methodological instructions for the implementation of the economic justification of the project for the development of software for the diploma thesis of the educational and qualification level "bachelor" for students of the specialties 121 - Software Engineering, 122 - Computer Science, 126 - Information Systems and Technologies in the Field of Knowledge 12 - Information Technologies. // Compiled by Moskalenko V. V., Shmatko O. V., Fonta N. G. - Kharkiv: NTU "KhPI" - 2022 - 35 p.
34. [Web Dean's Office - ACS "VNZ" (osvita.net)](https://vuz.osvita.net/veb-dekanat-main/)// https://vuz.osvita.net/veb-dekanat-main/, 02.06.2024.