Features of automation of the university schedule management process through the «Class Schedule WEB application»

Anna Firer[1], *Anatoly* Kopeikin[1], and *Igor* Khramov[2*]

[1]Lesosibirskij Pedagogical Institute – branch of Siberian Federal University, Pobedy Street. 42, Lesosibirsk, Russia
[2]Siberian Federal University, Svobodny Ave. 79, Krasnoyarsk, Russia

Abstract. An important direction in the management of an educational organization is the creation of optimal conditions for the academic work of students and teaching staff throughout the academic year, as well as the effective and uniform use of the university's classroom fund in compliance with all necessary rules and requirements, including sanitary and hygienic. Using Web applications to automate class schedules will significantly reduce the labor required to create and maintain class schedules. This work is aimed at researching the specifics of automating university schedule management through the Class Schedule Web application. Taking into account the identified limitations: a large number of educational programs, classrooms, teachers (including part-time students), academic groups of students, the work of specialized laboratories, etc. This system should facilitate the process of creating a schedule and keeping it up to date, reduce the likelihood of errors and provide quick access to schedule information for all interested users.

1 Introduction

The task of scheduling training sessions, including using automated scheduling programs, has been studied for many years [1-5] and is characterized by its multi-criteria nature, the complexity of limited time use, and the need to take into account many factors: from the presence of a large number of educational programs, classrooms, teachers (including part-time students), academic groups of students, the work of specialized laboratories, etc. is up to the individual wishes of the teachers.

When designing the «Class Schedule web application», the following tasks are defined:

- based on the domain analysis, identify the requirements for the web application and develop system models;
- to justify the choice of the architecture of the web application and the tools for its development;
- develop the server and client parts of the system;
- integrate the server and client parts of the system with the created database;

* Corresponding author: igor.07.06@mail.ru

- provide prompt access to the information of the web application for all participants in the educational process.

Potential users of the automated Lesson Schedule information system are unauthorized users (students and teachers) and authorized users (system administrator, specialist of the educational department (schedule manager)) [6].

Automating the process of creating and using the timetable through the Timetable Web application will significantly increase the efficiency of the educational organization, improve the management of the educational process and meet the needs of all participants in the educational process [7, 8].

2 Methods

In this scientific work, methods of observation and generalization, comparative analysis, methods of system decomposition, mathematical modeling, structural modeling, and interviewing were used to solve the tasks.

The design of the Lesson Schedule web application begins with a detailed analysis and a clear definition of requirements. At this stage, functional and non-functional requirements are being determined based on the study of existing methods and processes for compiling and editing the curriculum.

A lot of preliminary work is being done to schedule classes:

a) data collection:

- information about the academic workload of teachers (full name, name of disciplines);
- information about study groups (group code, student name list);
- information about available audiences;
- schedule of the educational process (start and end dates of the semester, dates of intermediate and final state attestation);
- study plans (serve as the basis for making schedules).

b) publication of schedules:

- the finished schedule is formed in tables, which are then posted on information resources.

c) schedule coordination:

- schedule changes according to current requirements.

The most important criteria for the quality of a software product include: user-friendly interface, flexibility of configuration, efficiency of the search algorithm, and reporting functions. [9]

The web application being developed should perform the following functions:

a) schedule compilation and editing:

- entering information about the name of the educational program, academic discipline, teachers, study groups, classrooms, and class times;
- real-time schedule editing and updating.

b) user management:

- registration and authentication of users (administrators, dispatchers);
- differentiation of access rights depending on user roles.

c) the ability to view schedules on various devices for users:

- access to the current class schedule at any time and from any device (computers, mobile phones, tablets);
- schedule display according to the selected filters.

d) authentication and authorization:

- implementation of authentication mechanisms to restrict access to the server side of the system only to authorized users.

Non-functional requirements determine the overall characteristics of a system, such as performance, security, and usability. The analysis of sources [10, 11] allowed us to identify the following requirements:

a) performance:

- the maximum acceptable response time for web applications is considered to be 5 seconds;
- the system must support simultaneous operation of a large number of users without reducing performance.

b) reliability and fault tolerance:

- automatic data backup should be performed daily, and the time to restore data from the backup should not exceed 1 hour. Such measures ensure data protection and recovery in the event of a failure.
- the system must correctly handle all errors and exceptions, ensuring continuous operation and logging of all incidents for further analysis.

c) ease of use

- the user interface should be intuitive and accessible, with a minimum number of clicks to perform basic operations.;
- the training time for users (students, teachers, administration) should be minimal, not exceeding 2 hours for each type of user.

d) protection from attacks:

- the application of protection measures against common vulnerabilities, such as the use of parameterized queries to prevent SQL injections and XSS attacks.;
- the architecture of the automated Timetable information system should ensure the flexibility, scalability, reliability and security of the system.

3 Results and discussion

The choice of technologies and tools plays a key role in the development of the automated Lesson Schedule information system. The main selection criteria are performance, scalability, ease of development, and community support. An important basis for an automated class scheduling system is the client-server architecture, which aims to ensure interaction between all components through a network.

The client/server model is an application development architecture designed to separate the data representation layer from its processing and storage. Using a client-server architecture allows you to achieve the following goals:

a) Separation of responsibilities. The client side is responsible for displaying schedules and interacting with the user, while the server side processes requests, executes business logic, and manages data. This separation allows you to effectively organize the operation of the application and provide users with the necessary functionality.

b) Scalability and performance. Dividing the system into client and server parts allows each of them to scale independently, ensuring high performance even with a large number of users. It also contributes to the fault tolerance of the system.

c) Data security and protection. The server side is responsible for data protection, including user authentication, access control, and data encryption. This ensures reliable protection of confidential information and prevents unauthorized access.

The principles of (clean Architecture) are used to create modular, flexible and easily supported systems [12]: independence from frameworks:

- the application code is not tied to specific technologies, which provides flexibility and the ability to easily switch to other tools;
- ease of testing: the architecture makes it easy to test each component of the system;
- independence from the user interface and database;
- allows to change the interface and type of database without affecting the business logic.

The system components interact through well-defined interfaces and protocols:

a) client-server interaction: the web interface (client) sends requests to the server via HTTP/HTTPS. The server processes requests, executes business logic, and interacts with the database;

b) API and microservices: using RESTful API to exchange data between client and server. The ability to split the system into microservices to increase flexibility and scalability.

When developing the client side of the system, the React JS library was used, which has high performance and a component-based approach.

The ER model was used to create the database of the Class Schedule web application. As part of the design, the following entities and their interrelationships are defined:

1) Entities:

- teacher;
- items;
- group;
- subgroups;
- schedule;
- type of occupation;
- audience;
- building;
- users;
- changes;
- roles.

2) Connections between entities:

- the teacher teaches several subjects;
- a group of students is studying several subjects;
- the group is divided into subgroups;
- the schedule contains changes that can be made by several users;
- classes are held in different classrooms;
- the classrooms are located in different buildings;
- classes have a specific type.

System interaction models are two types of interaction diagrams – sequence diagrams and communication diagrams, which are often used to visualize and analyze data flows. [13]. Figure 1 and Figure 2 show two main models of interaction: the interaction of the student with the system and the interaction of the schedule specialist with the AIS "Lesson Schedule".

The student requests and receives the schedule through the User Interface (UI). This process contains several key steps:

- request initiation: the student uses the UI to send a request to view the schedule;
- request processing on the server: the request is sent to the server, which executes the corresponding SQL query to extract data from the database;
- data return: Schedule data is returned from the server via the UI back to the student.

This model allows you to understand how data is transferred from the database to the end user and what steps are included in this process.

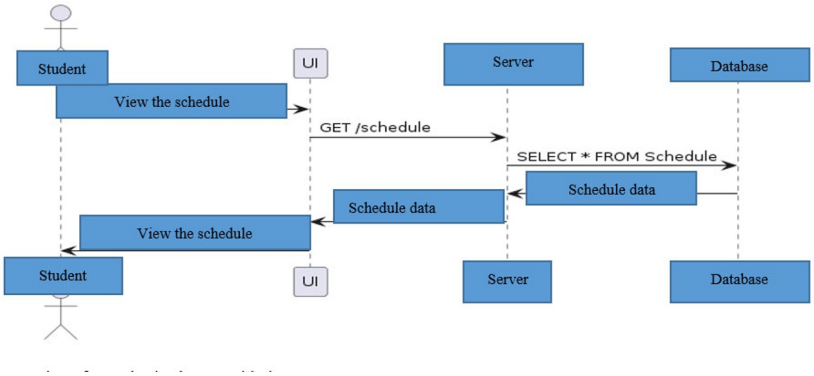

Fig. 1. Interaction of unauthorized users with the system.

Figure 2 shows the process of adding a new schedule by a schedule specialist. The addition process includes the following steps:

- initiation of the addition: the dispatcher sends the data of the new schedule through the UI of the administrative part;
- data processing on the server: The UI sends a POST request to the server, which executes an SQL query to add data to the database;
- getting the result: the server returns the result of the operation to the UI, informing the dispatcher about the success or failure of adding the schedule using pop-up notifications;
- user notification: The server sends a notification about the addition of a new schedule to the notification system, which then notifies the relevant users.

The dispatcher's interaction model with the schedule allows you to understand the process of making changes to the system and their impact on other components and users of the system.

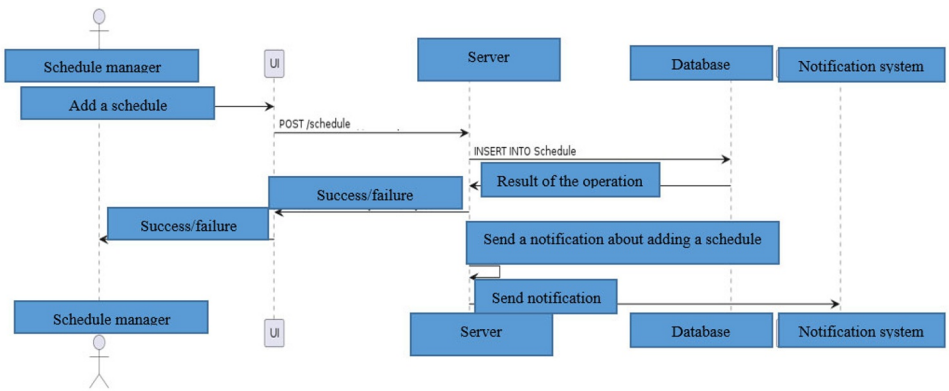

Fig. 2. Interaction of an authorized user with the system.

Sequence diagrams show the interaction with the system, which is an important tool for analyzing and designing the Schedule web application. They allow you to clearly present the sequence of actions and data exchange between the various participants and components of the system, which contributes to a more efficient development of a web application.

The server part of the system uses the Python programming language, which is characterized by relative simplicity and wide support for a variety of frameworks and tools [14]. As a web framework, Flask is flexible and allows for a quick start to development, and it also integrates seamlessly with various libraries and tools.

The MySQL database management system, characterized by high reliability and performance, is used as data storage (information about users, study groups, classrooms, etc.) [6, 13-16].

The Git version control system is used for source code management and collaboration. The Gitflow toolkit provides convenient introduction of new functions, correction and structuring of errors and releases, helps to manage versions and coordinate team work [6, 17].

4 Conclusion

The system was designed and developed taking into account modern requirements for web applications using available technologies and tools such as Python, Flask, React and JavaScript. In addition, the use of the genetic algorithm has made it possible to create an effective tool for automated management of the timetable process in an educational institution. The designed and implemented system provides for the possibility of interaction between the system and the user, which facilitates the process of creating and coordinating schedules, reduces the likelihood of errors and provides quick access to relevant information for all participants in the educational process.

References

1. D.A. Aganina, Problems of the automated schedule of the educational process. Young scientist **42** (2018)
2. R.I. Khabipov, *Automation of scheduling* (KFU Publishing House, Kazan, 2018)
3. T.A. Shishkanova, Algorithm for optimizing the academic schedule at a universityю Mining Information and Analytical Bulletin (scientific and Technical journal) **1** (2015)
4. T.S. Babkina, Business Informatics **1(3),** 21-28 (2008)
5. T.V. Sirkin, A.P. Chernyshova, P.A. Martynov, A.D. Morozov, Young Scientist **27(317)** 24-27 (2020)
6. A.V. Firer, E.V. Kirgizova, A.A. Kopeikin, P.A. Shelkunov, Modern high-tech technologies **7** 61-65 (2024)
7. A. Laure, *Web API Design* (O'Reilly Media, Sebastopol, 2020)
8. K. Vigers, *Development of software requirements* (BHV-Petersburg, St. Petersburg, 2014)
9. I.V. Bobrenko, Development of a program for automated scheduling at school based on a genetic algorithm in: *Open city scientific and practical conference "Engineers of the Future": collection of works by winners and prize-winners,* Moscow, April 18-20, 2019 (Publishing House of the Federal State Budgetary Educational Institution of Higher Professional Education "NIU "MEI", Moscow, 2020), pp. 476-477
10. E.A. Kaliberda, I.V. Fedotova, Bulletin of the Astrakhan State Technical University. Series: Management, Computer Science and Informatics **2(110),** 158-163 (2012)
11. A.V. Ostroukh, *Designing information systems* (Lan Publishing House, St. Petersburg, 2021)
12. R. Martin, *Pure Architecture. The Art of software development* (Peter, St. Petersburg, 2021)
13. L.A. Matsyashek, *Analysis and design of information systems using UML 2.0* (Williams, Moscow, 2008)

14. V.A. Dronov, *Django 3.0 The practice of creating websites in Python* (BHV-Petersburg, St. Petersburg, 2021)
15. R.D. Muller, *Database design and UML* (LORI Publ., Moscow, 2002)
16. S.V. Nazarov, *Architecture and design of software systems* (INFRA-M, Moscow, 2013)
17. I.V. Nikitin, The Diary of Science **3,** 3 (2013)