International Journal
Information Technology Applications (ITA)
Volume 15, Number 1, July 2026

MANAGEMENT OF DOCTOR ON-CALL SERVICES

Kristián Štefanka

Abstract:

The article focuses on the design, implementation, and evaluation of a web-based system for doctor shift scheduling. The aim was to create an efficient and user-friendly application that simplifies the scheduling process and reduces administrative workload. The system enables doctors to specify their availability preferences through an interactive calendar, while schedules are generated automatically based on predefined rules. The results show a significant improvement in scheduling efficiency, reduction of errors, and increased user satisfaction compared to traditional Excel-based approaches.

Keywords:

Shift scheduling, healthcare systems, web application, PHP, MySQL, FullCalendar, heuristic algorithm, optimization.

Introduction

The process of scheduling medical staff in healthcare institutions is a complex and demanding task that requires careful coordination of multiple factors [7]. Hospitals operate continuously, and therefore it is necessary to ensure that sufficient medical personnel are available at all times. The quality of scheduling has a direct impact on patient care, employee satisfaction, and the overall efficiency of healthcare services.

In practice, many healthcare institutions still rely on traditional scheduling methods, such as paper-based plans or spreadsheet applications like Microsoft Excel. While these approaches may be sufficient for small teams, they become increasingly inefficient and error-prone as the number of employees and scheduling constraints grows. Manual scheduling requires significant time and effort, and it is difficult to ensure that all constraints are consistently respected.

Modern information technologies provide new opportunities for improving this process. Web-based applications allow centralized data management, real-time updates, and accessibility from multiple devices. These systems can incorporate advanced algorithms that automate scheduling and reduce the likelihood of human error.

The main objective of this paper is to design and implement a web-based system for doctor shift scheduling. The system enables doctors to specify their availability preferences and allows administrators to generate schedules automatically. The solution aims to improve efficiency, reduce administrative workload, and ensure fair distribution of shifts among staff members.

1 Theoretical Background – Doctor Shift Scheduling

The process of scheduling doctor shifts in healthcare institutions represents a complex organizational and optimization problem that directly affects the quality of patient care, staff satisfaction, and the overall efficiency of medical services. Unlike standard workforce planning,

hospital scheduling must ensure continuous operation, which requires careful coordination of personnel across different time periods, departments, and medical specializations.

In modern healthcare environments, scheduling is influenced by a wide range of factors. These include the availability of doctors, their qualifications and specializations, legal regulations concerning working hours, and the need to maintain a balanced workload among staff members. In addition, personal preferences of doctors play an important role, as they contribute to employee satisfaction and long-term sustainability of the workforce.

One of the fundamental challenges in shift scheduling is the need to satisfy multiple constraints simultaneously. These constraints may include maximum working hours per week, mandatory rest periods between shifts, and the requirement to have a sufficient number of doctors present at all times. The complexity of these requirements makes manual scheduling increasingly difficult, especially in larger departments.

Traditionally, scheduling has been performed using paper-based systems or spreadsheet applications such as Microsoft Excel. While these approaches are simple and widely accessible, they are often inefficient and prone to human error. Manual updates, lack of automation, and difficulty in enforcing constraints lead to inconsistencies and increased administrative workload.

From a theoretical perspective, doctor shift scheduling can be classified as a combinatorial optimization problem [7]. The goal is to assign a set of doctors to a set of shifts in such a way that all constraints are satisfied while optimizing certain criteria, such as fairness or workload distribution. Exact optimization methods are often computationally expensive, which is why heuristic approaches are commonly used in practice.

Modern scheduling systems aim to address these challenges by utilizing digital tools and algorithms. Web-based applications provide centralized data storage, real-time updates, and improved accessibility. By integrating user preferences and automated decision-making processes, these systems significantly enhance the efficiency and reliability of scheduling.

2 System Design and Scheduling Approach

The design of the proposed system focuses on creating an efficient and user-friendly solution for managing doctor shift schedules. The main objective was to develop a system that allows intuitive interaction for users while ensuring accurate and reliable scheduling results.

The system is based on a web application architecture that enables centralized access for all users. Doctors can log into the system and specify their availability preferences using an interactive calendar. These preferences are categorized into three main states: preferred (“want to work”), unavailable (“cannot work”), and neutral (“can work”). This simple classification allows the system to process user input efficiently while maintaining flexibility.

The scheduling process is designed to minimize manual intervention. Administrators are responsible for overseeing the system and generating schedules, but the majority of the decision-making is handled automatically by the algorithm. This approach reduces the risk of human error and significantly speeds up the scheduling process.

A key aspect of the system design is the implementation of user roles. Two primary roles are defined: doctor and administrator. Doctors have limited access and can only manage their own preferences, while administrators have full control over the system, including user management and schedule generation. This separation ensures security and prevents unauthorized modifications. The overall system architecture is illustrated in (Fig.1).

Another important component is the database structure, which stores all relevant data, including user information, preferences, and generated schedules. The database is designed to maintain consistency and integrity of data, ensuring that all scheduling operations are performed correctly. The database structure of the system is illustrated in (Fig.2).

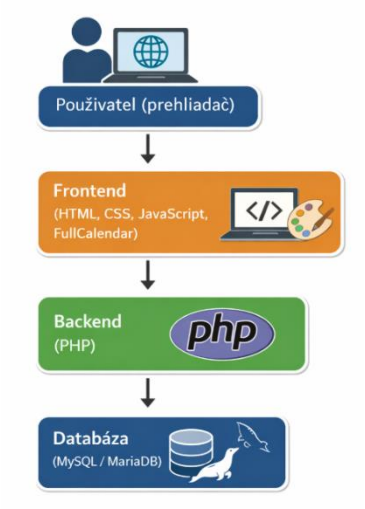

Fig.1. System architecture of the web-based scheduling application.

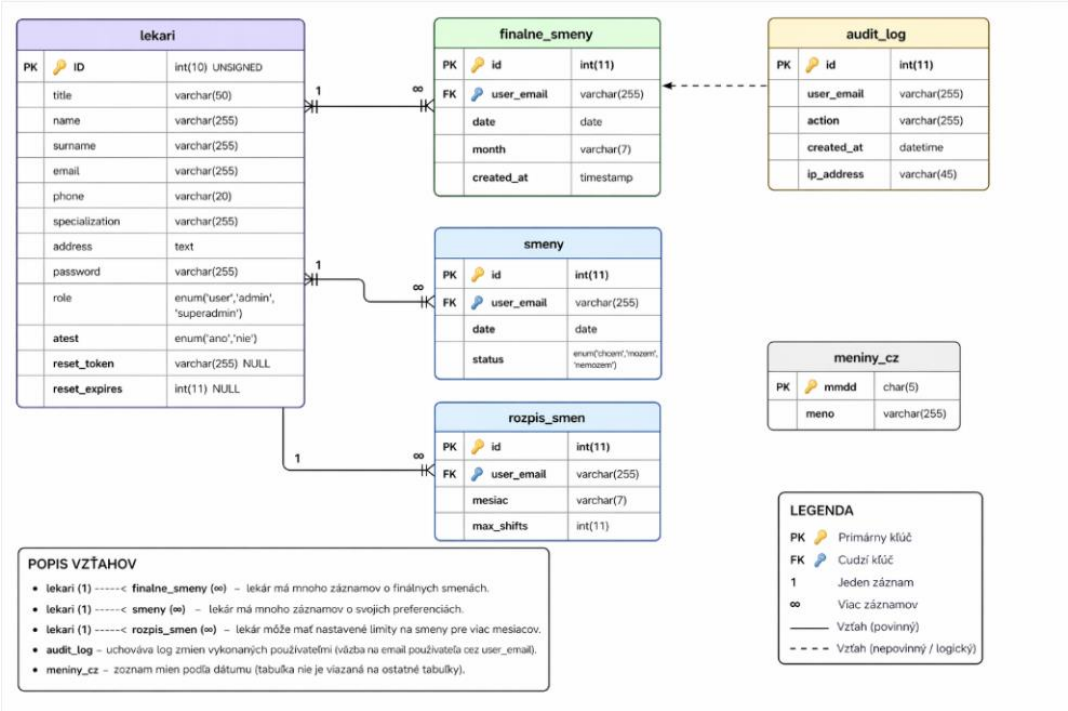

Fig.2. Entity-relationship diagram of the system database.

The scheduling approach is based on a heuristic algorithm [7] that prioritizes simplicity and ef- ficiency. Instead of searching for an optimal solution, the algorithm generates a feasible solution that satisfies all constraints and provides a fair distribution of shifts. This method is particularly suitable for real-world applications, where speed and reliability are more important than theoretical optimality. The scheduling process is illustrated in (Fig.4).

3 Tools Used and Implementation

The implementation of the system is based on widely used web technologies that ensure reliability, scalability, and ease of maintenance. The selection of these technologies was guided by their compatibility, performance, and availability.

The backend of the application is implemented using the PHP programming language [2], which handles server-side logic, processes user requests, and communicates with the database. PHP was chosen due to its simplicity, widespread support, and strong integration with web technologies.

The database layer is managed using MySQL [3], a relational database system that provides efficient data storage and retrieval. The database is designed to support multiple users and ensure data consistency even under concurrent access.

On the client side, the system uses HTML and CSS for structure and design, while JavaScript [4] provides interactivity. The Bootstrap framework [5] is used to create a responsive user interface that adapts to different screen sizes, ensuring usability on both desktop and mobile devices.

A key component of the system is the FullCalendar library [1], which provides an interactive calendar interface. This tool allows users to easily select and modify their availability preferences, offering a visual representation of scheduling data. User interactions and system functionality are illustrated in (Fig.3).

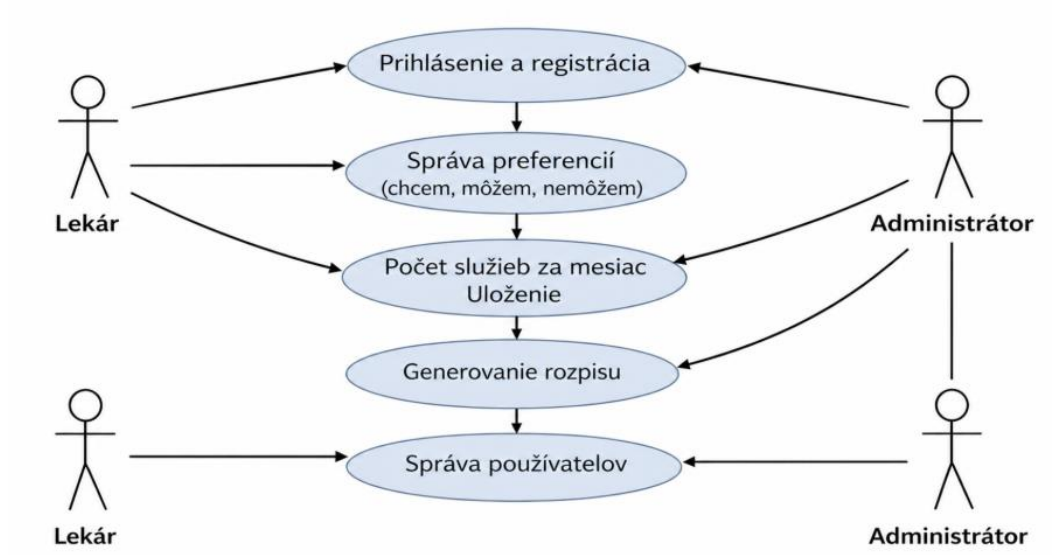

Fig.3. Use case diagram illustrating interactions between doctor and administrator.

Communication between the frontend and backend is implemented using AJAX [6], which enables asynchronous data exchange without reloading the page. This significantly improves user experience and system responsiveness.

The scheduling algorithm is implemented on the server side and processes data stored in the database. It evaluates user preferences and applies predefined rules to generate a schedule. The results are then stored and displayed to users in real time.

In addition to scheduling, the system also includes functionality for exporting data to Excel format. This feature is implemented using a dedicated library and allows administrators to generate reports for further analysis or archival purposes.

An example of the user interface is shown in (Fig.5).

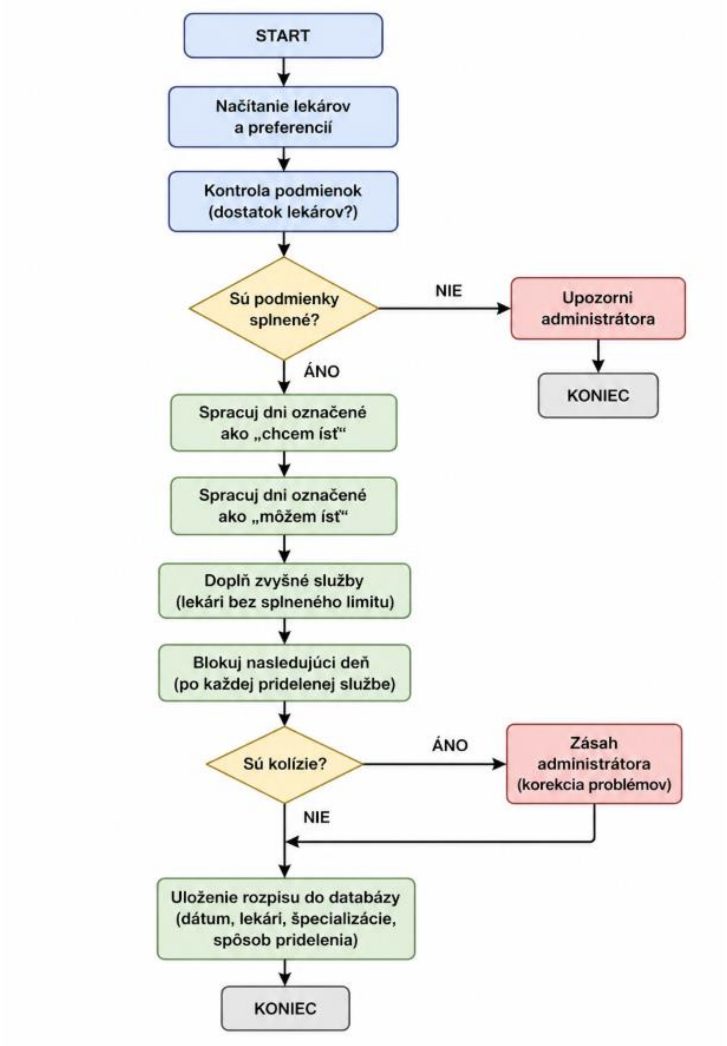

Fig.4. Flowchart of the scheduling algorithm.

4 Analysis of Results

The evaluation of the system represents a crucial part of the development process, as it provides deeper insights into its performance, reliability, and overall effectiveness in real-world conditions. The main objective of the analysis was to determine whether the system meets its intended goals, particularly in terms of efficiency, accuracy, and usability, and to compare its performance with traditional scheduling methods commonly used in healthcare institutions.

The evaluation was conducted using several test scenarios with varying numbers of users and different scheduling conditions. These scenarios were designed to simulate real-world usage, ranging from small departments with a limited number of doctors to larger environments with higher complexity. Key performance indicators included scheduling time, number of errors, system responsiveness, and user satisfaction. These metrics provide a comprehensive overview of the system's capabilities and allow for objective comparison with existing approaches.

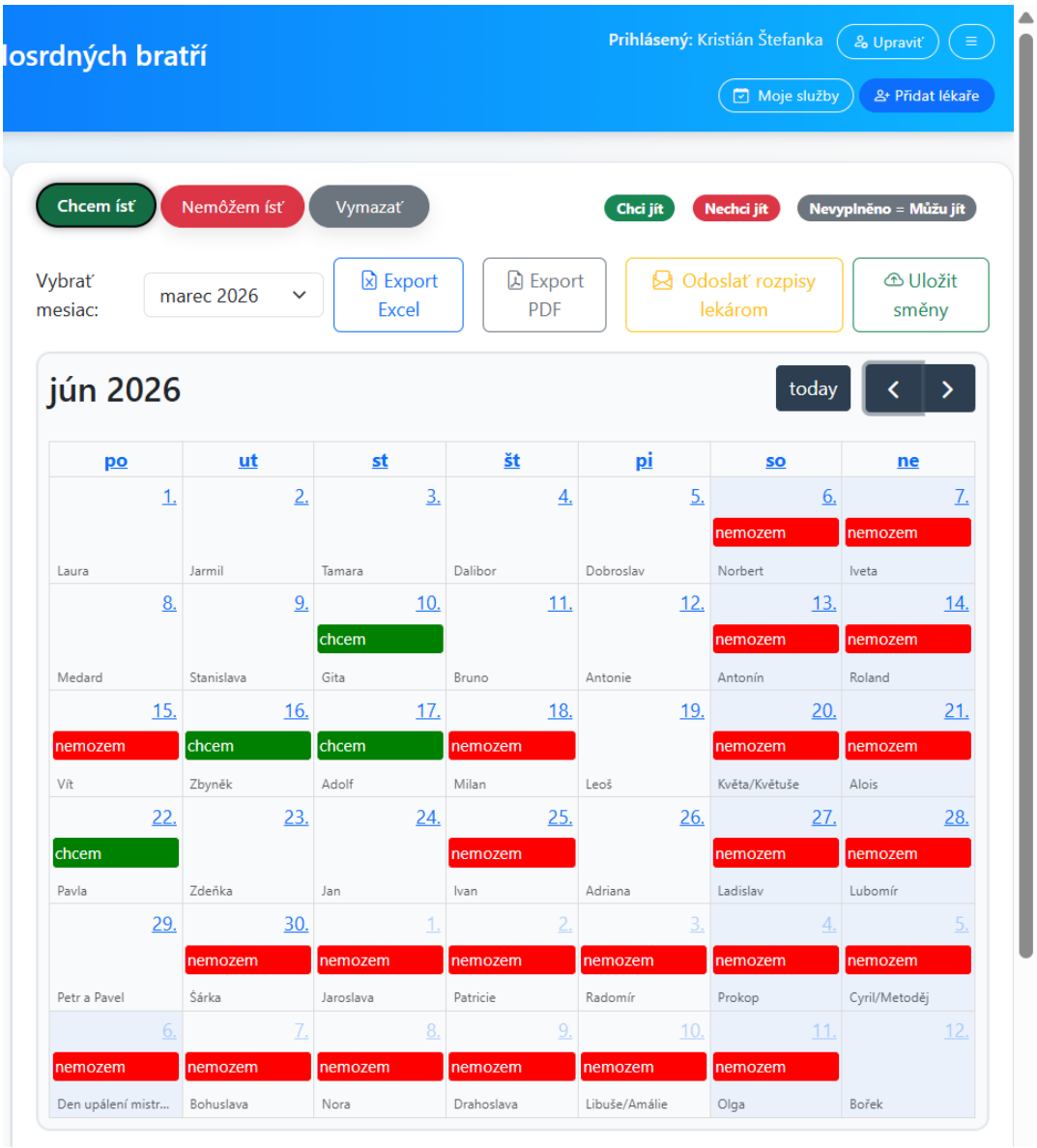

Fig.5. Interactive calendar for selecting doctor availability preferences.

The results showed a significant improvement compared to traditional Excel-based scheduling. In manual scenarios, the process of creating a schedule typically required several hours of work, especially when taking into account multiple constraints and preferences. In contrast, the proposed system was able to generate a complete schedule within a few seconds. This represents a substantial reduction in time and demonstrates the effectiveness of the automated approach. At the same time, the number of errors decreased significantly due to the implementation of automatic constraint validation, which prevents invalid assignments such as overlapping shifts or insufficient rest periods.

Another important aspect of the analysis was the evaluation of user experience. Feedback from users indicated that the system is intuitive and easy to use, even for individuals with limited technical experience. The visual representation of data through the interactive calendar interface was particularly appreciated, as it allows users to quickly understand and modify their preferences. The ability to input availability with a single click and immediately see the results of scheduling was identified as a major advantage over traditional methods, which often require manual updates and recalculations.

In addition to usability, the system was also evaluated in terms of stability and consistency. During testing, no critical failures or data inconsistencies were observed, which confirms the reliability of the implemented solution. The system maintained correct behavior even when multiple users interacted with it simultaneously, demonstrating its ability to handle concurrent access.

The system also demonstrated good scalability. Even with a higher number of users and increased complexity of scheduling requirements, the performance remained stable, and the scheduling process was completed within acceptable time limits. This confirms that the system is suitable not only for smaller departments but also for larger healthcare institutions where efficient scheduling is essential.

Furthermore, the evaluation included a comparison of qualitative aspects, such as transparency and fairness in shift allocation. The automated algorithm ensures that scheduling decisions are based on clearly defined rules and user preferences, which reduces the likelihood of subjective bias. This contributes to a more transparent and equitable distribution of shifts among doctors.

Overall, the analysis confirms that the proposed solution provides a significant improvement in efficiency, accuracy, and usability compared to traditional methods. The system not only reduces the time required for scheduling but also minimizes errors, improves user satisfaction, and provides a reliable and scalable platform for managing doctor shifts. These results highlight the potential of modern web-based applications to transform administrative processes in healthcare environments.

5 Identification of Problematic Areas

Despite the positive results, several limitations of the system were identified during testing and evaluation. One of the main limitations is the use of a heuristic algorithm, which does not guarantee an optimal solution. While the generated schedules satisfy all defined constraints and are suitable for practical use, it is possible that alternative solutions exist that could provide a more balanced distribution of shifts among doctors. This limitation is typical for heuristic approaches, which prioritize speed and simplicity over mathematical optimality. In more complex scenarios, especially with a higher number of constraints and users, the algorithm may produce solutions that are acceptable but not necessarily the most efficient or fair in the long term.

Another limitation is the lack of integration with external systems, such as hospital information systems, personnel management systems, or attendance tracking tools. In real-world healthcare environments, these systems are often interconnected, and the absence of integration may require redundant data entry or manual synchronization. Incorporating such integrations would significantly improve data consistency, reduce administrative workload, and enable more comprehensive management of hospital operations. For example, automatic synchronization with attendance systems could ensure that actual worked shifts are accurately recorded and reflected in future scheduling.

The current version of the system is primarily designed for web use, and although it is responsive and accessible on mobile devices through a browser, it does not include a dedicated mobile application. A native mobile application could provide a more optimized user experience, including faster interaction, offline access, and push notifications. Given that healthcare professionals often rely on mobile devices during their daily work, the absence of a mobile-specific solution may limit the system's usability in certain situations.

From a performance perspective, the system performs well under moderate load; however, its scalability has not been extensively tested in large-scale environments with hundreds of users and complex scheduling requirements. In such cases, performance optimization techniques, such as database indexing, caching mechanisms, or distributed system architecture, may be necessary to maintain efficiency and responsiveness.

Future improvements may also include the implementation of advanced optimization techniques, such as artificial intelligence or machine learning algorithms. These approaches could analyze historical scheduling data, identify patterns, and predict optimal staffing requirements. For example, machine learning models could learn from previous schedules and suggest improvements based on past conflicts, workload distribution, or user preferences. Such enhancements could significantly increase the quality of generated schedules and reduce the need for manual adjustments.

Additional features, such as notification systems or automated reminders, could further enhance the user experience and ensure better communication among staff members. Notifications could inform doctors about newly assigned shifts, changes in schedules, or upcoming duties, thereby reducing the risk of missed shifts or misunderstandings. Furthermore, implementing a feedback mechanism would allow users to report issues or suggest improvements, contributing to the continuous development of the system.

In conclusion, while the system provides a solid and functional foundation for efficient doctor shift scheduling, there are multiple opportunities for further development and improvement. Addressing the identified limitations and incorporating advanced features would enhance the system's robustness, scalability, and overall usability, making it more suitable for deployment in real-world healthcare environments.

Conclusion

The designed and implemented web-based system for doctor shift scheduling proved to be an effective solution for addressing the complex and time-consuming process of workforce planning in healthcare institutions. The system successfully integrates modern web technologies with a practical scheduling approach, providing a reliable platform that supports both administrators and medical staff. Particular emphasis was placed on improving efficiency, reducing human error, and ensuring fair distribution of shifts, which are key requirements in real-world hospital environments.

One of the main contributions of the system is the significant reduction in administrative workload. Traditional scheduling methods, especially those based on spreadsheets or manual planning, require considerable time and effort, often leading to inconsistencies and errors. In contrast, the proposed system automates the majority of scheduling tasks, allowing administrators to generate complete schedules within seconds. This not only increases productivity but also enables more frequent updates and adjustments in response to changing conditions.

The evaluation results confirmed that the system achieves a high level of accuracy and reliability. Automated constraint checking ensures that all generated schedules comply with predefined rules, such as rest periods, maximum workload, and availability constraints. As a result, the number of scheduling errors is significantly reduced compared to traditional methods. This contributes to safer and more efficient operation within healthcare institutions, where scheduling mistakes can have serious consequences.

Another important aspect of the system is its user-oriented design. The implementation of an interactive calendar interface allows doctors to easily input their availability preferences in a clear and intuitive manner. This not only simplifies user interaction but also increases engagement and satisfaction among staff members. The ability to visually interpret scheduling data and immediately see the results of changes represents a major improvement over static and less flexible traditional tools.

The system also demonstrated strong performance in terms of scalability and adaptability. It was able to handle different scheduling scenarios with varying numbers of users while maintaining stable performance and consistent results. This indicates that the solution is suitable not only for small departments but also for larger healthcare institutions with more complex requirements.

An additional benefit of the implemented system is the increased transparency and fairness in shift allocation. Since the scheduling process is based on clearly defined rules and user preferences, it minimizes the influence of subjective decision-making. This contributes to a more balanced distribution of workload among doctors and helps prevent conflicts within teams.

Despite these advantages, the system also highlights several areas for further improvement. The use of a heuristic algorithm, while efficient, does not guarantee an optimal solution in all cases. Future development could focus on incorporating advanced optimization methods, such as artificial intelligence or machine learning, to enhance decision-making and improve the overall quality of generated schedules. Furthermore, integration with external hospital systems and the development of a dedicated mobile application would significantly increase the system's practical usability.

In conclusion, the proposed system represents a modern and effective approach to doctor shift scheduling, addressing the limitations of traditional methods and demonstrating the potential of web-based solutions in healthcare management. The results confirm that digitalization and automation can significantly improve administrative processes, reduce errors, and enhance user experience. With further development and integration, the system has the potential to become a comprehensive tool for managing workforce scheduling in healthcare environments.

Acknowledgement

This work is part of the author's bachelor thesis, which is being prepared and will be defended in the current academic year.

I would like to express my sincere gratitude to my supervisor for his guidance, valuable advice, and continuous support throughout the development of this work.

I also thank the medical staff who participated in testing the system and provided useful feedback. The system was tested in the Hospital of the Merciful Brothers in Brno, where it is currently being gradually introduced into practical use, as no dedicated scheduling system was previously available.

References

[1] FullCalendar (2025). *JavaScript Event Calendar*.
Available at: https://fullcalendar.io/ (accessed January 10, 2026).

[2] PHP.net (2025). *PHP: Hypertext Preprocessor Documentation*.
Available at: https://www.php.net/docs.php (accessed January 10, 2026).

[3] Oracle (2025). *MySQL Documentation*.
Available at: https://dev.mysql.com/doc/ (accessed January 10, 2026).

[4] MDN Web Docs (2025). *JavaScript Guide*.
Available at: https://developer.mozilla.org/en-US/docs/Web/JavaScript (accessed January 10, 2026).

[5] Bootstrap (2025). *Bootstrap Documentation*.
Available at: https://getbootstrap.com/docs/ (accessed January 10, 2026).

[6] GeeksforGeeks (2025). *AJAX Introduction and Working*.
Available at: https://www.geeksforgeeks.org/ajax-introduction/ (accessed January 10, 2026).

[7] Nurse Scheduling Problem (2025). *Overview of scheduling in healthcare*.
Available at: https://en.wikipedia.org/wiki/Nurse_scheduling_problem (accessed January 10, 2026).

Authors

Kristián Štefanka
Faculty of Informatics
Pan-European University in Bratislava, Slovakia
xstefanka@paneurouni.com
Kristián Štefanka is a student of Applied Informatics at the Faculty of Informatics, Pan-European University in Bratislava. His academic interests include web application development, database systems, and optimization algorithms, particularly in healthcare process automation.