Vol. 6, No. 1, 2025, pp. 1-8
DOI: https://doi.org/10.29210/8815201

Contents lists available at Journal IICET

International Journal of Technology, Innovation and Humanities
ISSN: 2746-6434 (Electronic)

Journal homepage: http://journal.iicet.org/index.php/ijtih

Facility booking system: a key to unlocking organizational efficiency and effectiveness

Afsal Abdullah*), Sharifah Nurulhuda Tuan Mohd Yasin, Rozita Md. Ali
Politeknik Kuala Terengganu, Terengganu, Malaysia

Article Info

Article history:

Received Jun 12th, 2025
Revised Aug 20th, 2025
Accepted Aug 26th, 2025

Keywords:

Operational efficiency
Resource management
Web-based platform

ABSTRACT

Effective management of facilities plays a critical role in organizations' operational efficiency and effectiveness. Using manual systems for facility booking may result in inefficiencies, inaccuracies, and resource wastage. This article introduces a Facility Booking System (FBS) developed to improve the optimal usage of facilities, decrease expenditures, and enhance the overall performance of organizations. The FBS, which operates through a web-based platform using visual studio, PHP and Laragon Services, allows users to reserve facilities, oversee bookings, and monitor the utilization of facilities. The findings of the analysis support the FBS provided, exhibiting both advantages and obstacles related to the system. The findings indicate notable enhancements in facility usage, a decrease in booking mistakes, and an increased level of satisfaction among users. The article argues that implementing the FBS is pivotal in unlocking organizations' operational efficiency and effectiveness, concluding with recommendations for entities contemplating the adoption of similar systems.

Corresponding Author:

Afsal Abdullah,
Politeknik Kuala Terengganu
Email: nur.sal.22@gmail.com

Introduction

In today's dynamic business environment, organizations strive for maximum efficiency and effectiveness to remain competitive. A crucial aspect of this pursuit is optimizing the use of physical resources, including meeting rooms, conference halls, training spaces, and other facilities. This is where a Facility Booking System (FBS) emerges as a powerful tool, facilitating the smooth and efficient management of these resources. Therefore, knowledge related to database management systems is necessary; it offers a variety of techniques to store and retrieve data to enable the functions developed to meet the study's objective, which is to efficiently manage the application of UIDM services and facilities (Nordeen, 2022).

A Facility Booking System (FBS) is a software solution designed to automate and simplify the process of booking, scheduling, and managing organizational facilities. This is because database is a set of linked data that is carefully stored redundantly to best serve one or more applications to avoid duplication of orders (Dirgantara et al.). It provides a centralized platform for users to browse available spaces, reserve them for specific events or activities, and access real-time information about facility availability and usage. Through the internet, end users can access the database and perform specific processes according to their permission (Chassiakos & Sakellaropoulos, 2008).

Method

This article introduces a Facility Booking System (FBS) developed to improve the optimal usage of facilities, decrease expenditures, and enhance the overall performance of organizations. The FBS, which operates through a web-based platform using visual studio, PHP and Laragon Services, allows users to reserve facilities, oversee bookings, and monitor the utilization of facilities. The development of a Facility Booking System (FBS) referring to Agile Modeling method as a guideline to identify the problem statement, objective and solution to overcome the problem arise with manual facility booking process. Agile will enhance the whole system development process by following all the step-in agile modeling (Erickson et al., 2007). Therefore, this modelling will help to emerged the research for next process.

The system provides an interactive interface developed using Visual Code, ensuring a user-friendly experience for service applicants and UIDM administration. This program will help create and modify web projects for the supported programming languages (Johnson, 2019). The primary aim of this innovative system is to revolutionize the application and confirmation processes, offering a more efficient and systematic approach. The development of this system uses PHP and Laragon services to ensure that all system specifications are met and costly. Therefore, the Laragon program is the best choice for establishing a scientific study database; it is free and convenient (o'g'li & Rajabovich, 2023). PHP proposes a simple way to create dynamic web pages and enables complete control of the developed web, such as the storage process and access from the database (Nixon, 2021). PHP's economy, scalability, simplicity, and compatibility account for its popularity. An analysis of popular PHP programming topics is conducted (Mphanza & Mupala, 2019).

Figure 1 <Agile Model>
(Resource: https://images.app.goo.gl/RGAPQMeN5xmxsp2V6)

This Data Flow Diagram (DFD) illustrates the architectural design and information flow patterns of a service booking and management system, demonstrating the systematic movement of data between various user roles, processes, and data repositories within the organizational framework. The diagram employs standard DFD notation conventions, utilizing numbered processes, external entities, data flows, and data stores to represent the complete system architecture (Yourdon & Constantine, 1979). The system architecture is structured around a centralized login homepage (Process 1.0) that serves as the primary authentication gateway, facilitating role-based access control for three distinct user categories: Administrator, Head of Department, and Staff members. This hierarchical access model reflects contemporary principles of information systems security, where differential access privileges are assigned based on organizational roles and responsibilities (Whitman & Mattord, 2018).

The data flow architecture demonstrates a comprehensive user lifecycle management system, beginning with user registration (Process 3.0) and profile management (Process 2.0) functionalities. The bidirectional data flows between these processes and the User Information data store indicate the system's capability to maintain persistent user data while enabling dynamic profile updates. This design pattern is consistent with modern database-driven applications that require real-time data synchronization and integrity maintenance (Elmasri & Navathe, 2016). The core business logic is implemented through the service booking workflow (Process 4.0), which interfaces with a support application module (Process 5.0) to provide comprehensive service management capabilities. The data flow from the booking process to the Booking Information data store demonstrates the system's transaction processing capabilities, while the subsequent flows to the application checking process (Process 6.0) indicate an approval workflow mechanism typical of enterprise resource planning systems (Monk & Wagner, 2013).

The reporting subsystem (Process 7.0) represents a critical component for organizational decision-making, generating comprehensive reports from booking data and status information. The data flows indicate that this module can access information from multiple sources, including booking records and application status data, enabling multi-dimensional analysis and reporting capabilities. This design reflects best practices in management

information systems, where data aggregation and analysis tools provide strategic insights for organizational leadership (Laudon & Laudon, 2020).

The diagram reveals a sophisticated approval workflow where the Head of Department maintains oversight authority over support applications, while administrative personnel can verify application statuses independently. This multi-level approval process ensures appropriate governance and control mechanisms are embedded within the system architecture, aligning with organizational compliance requirements and audit trail maintenance (Romney & Steinbart, 2018).

The external entity relationships demonstrate clear separation of concerns between different organizational roles, with each user type having specific interaction patterns with the system. The Staff entity primarily engages with booking and support functions, while the Head of Department focuses on approval and oversight activities, and the Administrator maintains system-wide management capabilities. This role-based interaction model supports the principle of least privilege access control, minimizing potential security vulnerabilities while maintaining operational efficiency.

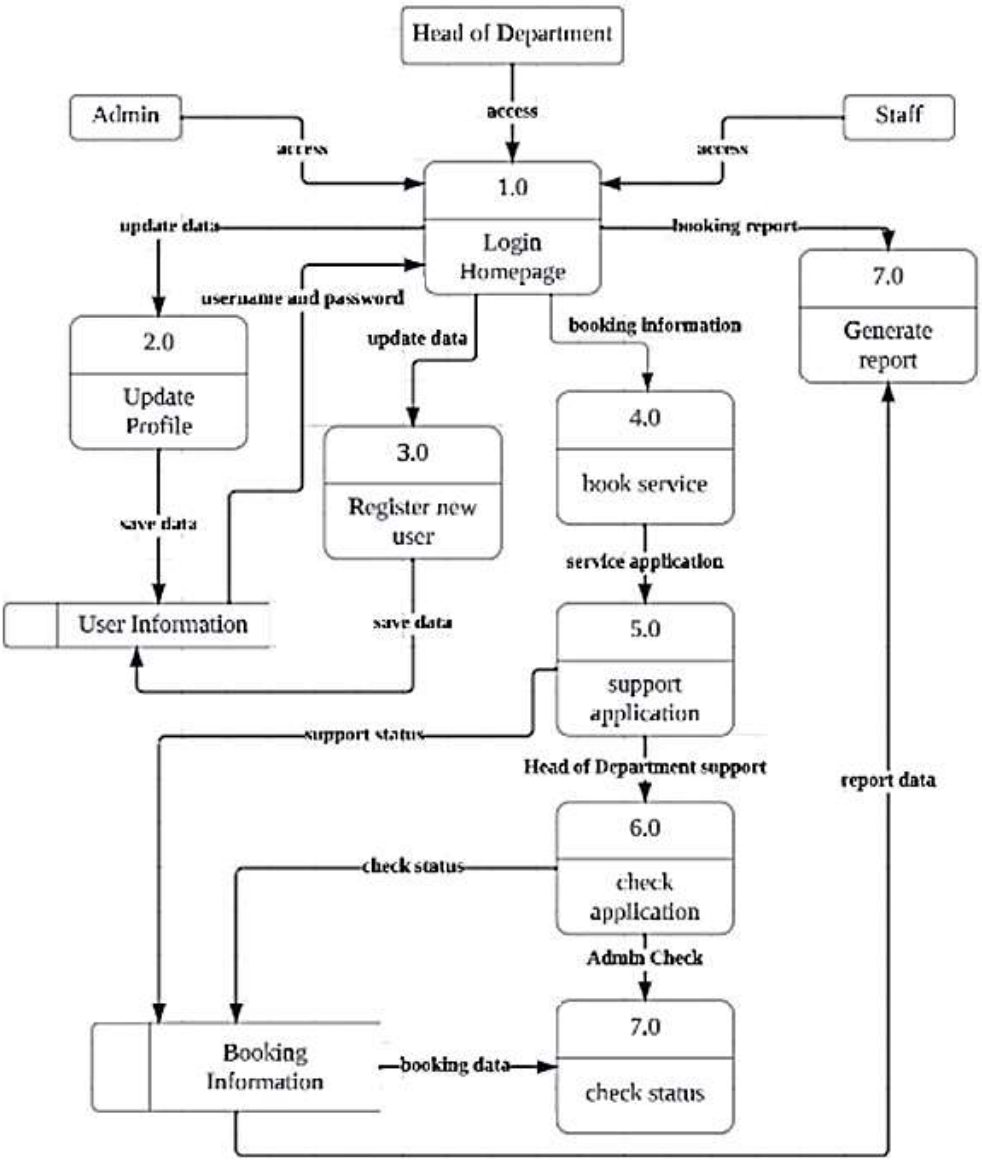

Figure 2 < Data Flow Diagram of the system >

This Entity Relationship Diagram (ERD) (Figure 3) represents the conceptual data model for a comprehensive application management system, illustrating the structural relationships and dependencies between core business entities within the organizational database architecture. The diagram employs standard ERD notation conventions,

utilizing entity boxes with attribute listings and relationship connectors to define the logical data structure (Chen, 1976). The central entity in this relational model is the "application" table, which serves as the primary business object containing critical application processing information including unique identifiers, user associations, temporal data, purpose specifications, digital signatures, and multi-level status tracking mechanisms. This entity demonstrates characteristics of a transactional system designed to manage complex approval workflows with comprehensive audit trail capabilities (Date, 2012). The inclusion of attributes such as app_support_status, app_admin_status, and app_support_note indicates a sophisticated multi-tier approval process that accommodates different organizational authorization levels.

The "user" entity represents the system's authentication and authorization foundation, containing comprehensive user profile information including identification credentials, contact details, departmental affiliations, and security credentials. The relationship between the user and application entities establishes a one-to-many cardinality, indicating that individual users can submit multiple applications while each application is associated with a single user account. This design pattern is consistent with standard user management practices in enterprise information systems (Silberschatz et al., 2019). The "department" entity provides organizational context and hierarchical structure to the system, containing departmental identification codes and descriptive information. The relationship between the user and department entities suggests a mandatory departmental affiliation for all system users, reflecting typical organizational structures where user permissions and workflow routing may be determined by departmental membership (Rob & Coronel, 2017).

The system incorporates two specialized sub-entities: "application_service" and "application_facility," both maintaining foreign key relationships to the primary application entity. The application_service entity manages service-related requests with temporal scheduling capabilities, including location specifications and time-bounded service delivery parameters. This entity structure suggests the system accommodates location-dependent services with specific scheduling requirements, typical of facility management or resource booking systems. The application_facility entity handles facility-related applications with detailed temporal scheduling attributes, including start and end time specifications, facility identification, and associated notes. The temporal granularity of this entity indicates sophisticated scheduling capabilities that can manage time-sensitive facility reservations or equipment bookings with conflict resolution mechanisms.

The relational architecture demonstrates third normal form compliance, with appropriate primary key definitions and foreign key constraints that ensure referential integrity across the database structure. The attribute naming conventions follow standard database design practices, utilizing descriptive prefixes that clearly identify entity ownership and facilitate query optimization (Ramakrishnan & Gehrke, 2016). The overall schema design reflects a comprehensive business application management system capable of handling multiple application types, complex approval workflows, and detailed scheduling requirements while maintaining data integrity and supporting sophisticated reporting capabilities. The inclusion of timestamp attributes (app_created_at, app_facility_date) enables temporal analysis and audit trail maintenance, essential components of enterprise-grade application management systems.

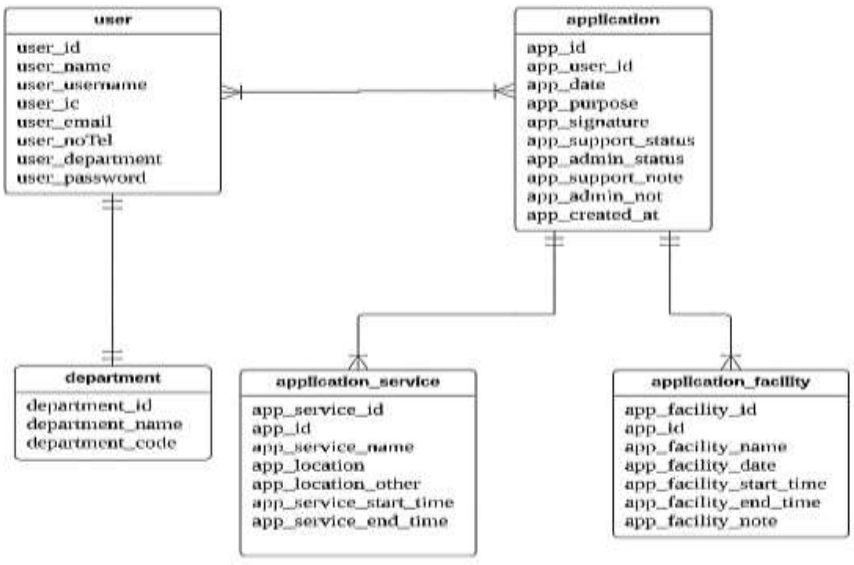

Figure 3< Entity Relationship Diagram of the system >

Results and Discussions

User Interface

Three main modules have been developed: Staff Module, Head of Department, and UIDM Admin. Functions for each module have been identified to ensure that the work procedures implemented are under the work process manual that has been set but improved in terms of access and efficient access to overcome the issue of duplicate bookings, loss of facility application forms and access to usage records during audits. Four functions are developed in this module, including the function of user registration, review and approval of applications and the display of booking reports referring to Figure 4.

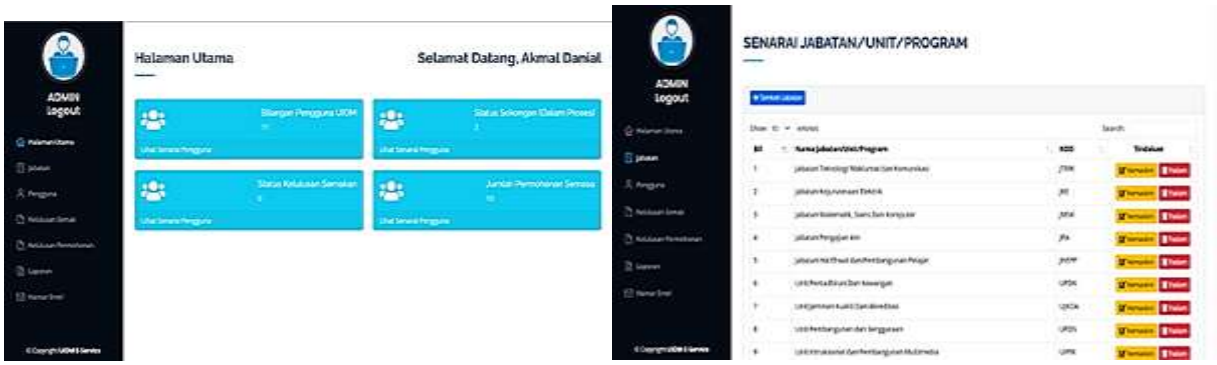

Figure 4 <Module: UIDM Admin>

Head of Department (HED) have a responsibility to approve or deny applications made by subordinates; this function allows a head of department to review the application. HED can monitor the activity of subordinates and their progress.

Figure 5 <Module: Head of Department>

Authorized users are allowed to apply for a facility provided by the UIDM department. The user needs to identify the required facility and choose the appropriate date. The user cannot complete the booking process if the date and service are already booked by other staff. Moreover, applicants can check the approval status of facility bookings.

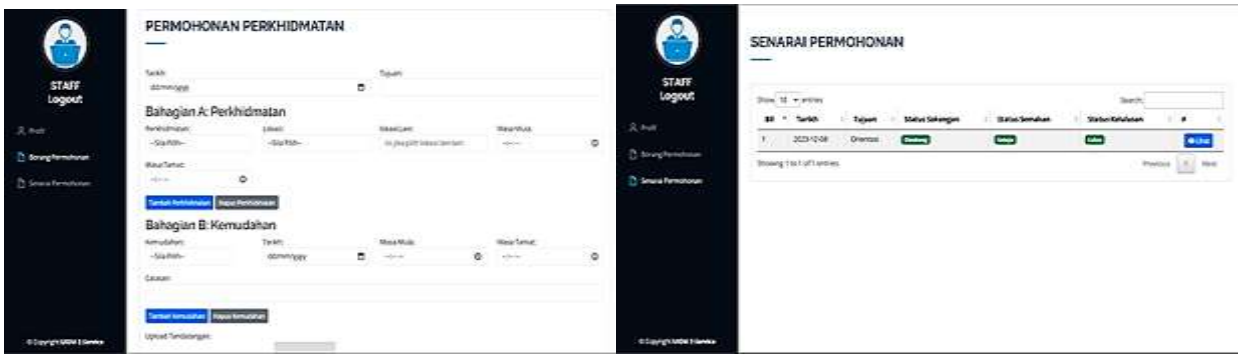

Figure 6 <Module: Staff>

User Acceptance Test
The UIDM admin has implemented the testing system to confirm that each process and procedure is accurate and appropriate to apply.

Figure 7 <Unit Testing Plan >

Test Case Name	Test Procedure	Pre-condition	Expected Result	(UIDM Admin)	Result (Pass/Fail)
Login	User is required to click Login button	none	User will go to several dashboards based on the user's role	Wan Hasbullah Bin Wan Abdul Halim	Pass
Register new user	Admin is required to click 'Daftar' button	none	Prompt notification successful register	Wan Hasbullah Bin Wan Abdul Halim	Pass
Update profile	User is required to click 'Kemaskini' button	none	Prompt notification successful update	Wan Hasbullah Bin Wan Abdul Halim	Pass
Book service	User is required to click 'Mohon' button	none	User will go to "Senarai permohonan" and status application pending	Wan Hasbullah Bin Wan Abdul Halim	Pass
Support application	HED is required to click 'Kemaskini' button	none	HED will go to "Semakan Permohonan" and support status is approved	Wan Hasbullah Bin Wan Abdul Halim	Pass
Checking application	Admin is required to click 'Kemaskini' button	none	Admin will go to "Kelulusan Permohonan" and check status approved	Wan Hasbullah Bin Wan Abdul Halim	Pass
Generate report	Admin required to choose yearly and monthly in dropdown list	none	Display all information for the selected monthly and yearly	Wan Hasbullah Bin Wan Abdul Halim	Pass
Email	Admin is required to click 'Send' button	none	Prompt notification successful email	Wan Hasbullah Bin Wan Abdul Halim	Pass

Conclusions

The findings of the analysis support the FBS provided, exhibiting both advantages and obstacles related to the system. The findings indicate notable enhancements in facility usage, a decrease in booking mistakes, and an increased level of satisfaction among users. The article argues that implementing the FBS is pivotal in unlocking organizations' operational efficiency and effectiveness, concluding with recommendations for entities contemplating the adoption of similar systems. The empirical findings presented in this analysis constitute a significant contribution to the body of knowledge regarding Facility Booking Systems (FBS) implementation and their impact on organizational operational performance. The documented improvements in facility utilization rates, reduction in booking errors, and enhanced user satisfaction metrics demonstrate the tangible benefits that technology-mediated resource management systems can deliver within organizational contexts (Brynjolfsson & Hitt, 2000). These results align with broader theoretical frameworks in information systems research that emphasize the transformative potential of digital technologies in optimizing resource allocation and improving operational efficiency (Melville et al., 2004).

Comparative studies examining FBS implementation across diverse organizational types, sizes, and cultural contexts would significantly enhance the generalizability of current findings. The heterogeneity of organizational environments suggests that implementation outcomes may vary substantially based on contextual factors such as organizational culture, technological infrastructure, and management support systems (Hofstede, 2001). Cross-organizational comparative research could identify critical success factors and implementation best practices that transcend organizational boundaries. Future investigations should incorporate more sophisticated methodological approaches, including mixed-methods designs that combine quantitative performance metrics with qualitative assessments of user experience and organizational change processes. Such approaches would provide deeper insights into the mechanisms through which FBS implementation generates operational improvements and user satisfaction enhancements (Creswell & Plano Clark, 2017). Additionally, the application of advanced analytical techniques, such as structural equation

modeling or machine learning algorithms, could reveal complex relationships between implementation factors and performance outcomes that are not apparent through traditional analytical approaches.

Acknowledgments

The authors would like to extend their sincere gratitude to Politeknik Kuala Terengganu, Unit Penyelidikan Inovasi dan Komersialan family and colleagues who have made significant contributions to various parts of this research endeavors.

References

Brynjolfsson, E., & Hitt, L. M. (2000). Beyond computation: Information technology, organizational transformation and business performance. Journal of Economic Perspectives, 14(4), 23-48.

Charntaweekhun, K., & Wangsiripitak, S. (2006). Visual programming using flowchart. 2006 International Symposium on Communications and Information Technologies. https://doi.org/10.1109/iscit.2006.339940

Chassiakos, A. P., & Sakellaropoulos, S. P. (2008). A web-based system for managing construction information. Advances in Engineering Software, 39(11), 865–876. https://doi.org/10.1016/j.advengsoft.2008.05.006

Chen, P. P. (1976). The entity-relationship model—toward a unified view of data. *ACM Transactions on Database Systems*, 1(1), 9-36.

Creswell, J. W., & Plano Clark, V. L. (2017). Designing and conducting mixed methods research (3rd ed.). Sage Publications.

Date, C. J. (2012). *An introduction to database systems* (8th ed.). Pearson.

Dirgantara, M. R., Syahputri, S., Hasibuan, A., & N, N. (n.d.). Pengenalan Database Management System (DBMS). Madani: Jurnal Ilmiah Multidisiplin. https://jurnal.penerbitdaarulhuda.my.id/index.php/MAJIM/article/view/411

Elmasri, R., & Navathe, S. B. (2016). Fundamentals of database systems (7th ed.). Pearson.

Erickson, J., Lyytinen, K., & Siau, K. (2007). Understanding Agile Software, Extreme Programming, and Agile Modeling. Research Issues in Systems Analysis and Design, Databases and Software Development. https://doi.org/10.4018/9781599049274.ch002

Hammer, M., & Champy, J. (2003). Reengineering the corporation: A manifesto for business revolution. HarperBusiness.

Hofstede, G. (2001). Culture's consequences: Comparing values, behaviors, institutions and organizations across nations (2nd ed.). Sage Publications.

Johnson, B. (2019). Visual studio code: End-to-end editing and debugging tools for web developers. Wiley.

Laudon, K. C., & Laudon, J. P. (2020). Management information systems: Managing the digital firm (16th ed.). Pearson.

Melville, N., Kraemer, K., & Gurbaxani, V. (2004). Information technology and organizational performance: An integrative model of IT business value. MIS Quarterly, 28(2), 283-322.

Monk, E., & Wagner, B. (2013). Concepts in enterprise resource planning (4th ed.). Course Technology.

Nixon, R. (2021). Learning PHP, MySQL & JavaScript. United State of America: O'Reilly Media, Inc.

Nordeen, A. (2022, July 19). Learn DBMS in 24 hours by Alex Nordeen (ebook) - read free for 30 days. Everand. https://www.everand.com/book/583196870/Learn-DBMS-in-24-Hours

O‘Roqov Asliddin Doniyor O‘G‘Li, & Mumin Babajanov Rajabovich. (2023). Analysis And Advantages of Laragon Software In Data Processing. Journal Of Innovation, Creativity and Art, 2(12), 137–142. Retrieved From Http://Jica.Innovascience.Uz/Index.Php/Jica/Article/View/435

Orlikowski, W. J., & Scott, S. V. (2008). Sociomateriality: Challenging the separation of technology, work and organization. Academy of Management Annals, 2(1), 433-474.

Porter, M. E., & Heppelmann, J. E. (2014). How smart, connected products are transforming competition. Harvard Business Review, 92(11), 64-88.

Ramakrishnan, R., & Gehrke, J. (2016). *Database management systems* (3rd ed.). McGraw-Hill.

Rob, P., & Coronel, C. (2017). *Database systems: Design, implementation, and management* (12th ed.). Cengage Learning.

Rogers, E. M. (2003). Diffusion of innovations (5th ed.). Free Press.

Romney, M. B., & Steinbart, P. J. (2018). Accounting information systems (14th ed.). Pearson.

Sambamurthy, V., Bharadwaj, A., & Grover, V. (2003). Shaping agility through digital options: Reconceptualizing the role of information technology in contemporary firms. MIS Quarterly, 27(2), 237-263.

Silberschatz, A., Galvin, P. B., & Gagne, G. (2019). *Database system concepts* (7th ed.). McGraw-Hill.

Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. MIS Quarterly, 27(3), 425-478.
Welling, L., & Thomson, L. (2003). PHP and MySQL Web development. Sams publishing.
Whitman, M. E., & Mattord, H. J. (2018). Principles of information security (6th ed.). Cengage Learning.
Yourdon, E., & Constantine, L. L. (1979). Structured design: Fundamentals of a discipline of computer program and systems design. Prentice-Hall.