Preprints.org

Article

Not peer-reviewed version

System Acquisition and Development of the Journal House Information System Using Bespoke and Prototype for Improve Connectivity on Integrity Services

Eva Nurlatifah , Raden Ibnu Huygenz Widodo * , Reza Fahlevi Herdiyanto , Rifqi Syekhi Marsaputra , Rifky Zaini Faroj

Posted Date: 23 July 2024

doi: 10.20944/preprints202407.1723.v1

Keywords: system acquisition; development; information system

Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 July 2024 doi:10.20944/preprints202407.1723.v1

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

System Acquisition and Development of the Journal House Information System Using Bespoke and Prototype for Improve Connectivity on Integrity Services

Eva Nurlatifah [1], Raden Ibnu Huygenz Widodo [2], Reza Fahlevi Herdiyanto [2], Rifqi Syekhi Marsaputra [2] and Rifky Zaini Faroj [2]

[1] UIN Sunan Gunung Djati Bandung, Jawa Barat, Indonesia; evanurlatifah@uinsgd.ac.id
[2] Informatics Department, UIN Sunan Gunung Djati Bandung, Jawa Barat, Indonesia; rfahlevih@gmail.com (R.F.H.); rifqisyekhi@gmail.com (R.S.M.); rifkyzainix@gmail.com (R.Z.F.)
* Correspondence: ibnuwdodo@gmail.com

Abstract: The UIN Sunan Gunung Djati Journal House Information System encounters significant usability challenges stemming from inadequate UI/UX design and sub-optimal integration across its components, resulting in diminished user efficiency and adoption rates. To tackle these issues effectively, a bespoke acquisition approach coupled with prototype development was implemented. This method facilitated a structured software development life cycle emphasizing iterative feedback loops and active stakeholder participation. By iteratively refining requirements and ensuring continuous alignment with user expectations, the prototype development process aimed to deliver a final system that not only meets but exceeds user needs. Rigorous testing, ongoing support, and a commitment to continuous improvement were pivotal in transforming the initial prototype into a robust and user-friendly system. This approach not only promises to enhance overall system effectiveness but also ensures sustained value delivery to stakeholders over the system's lifecycle.

Keywords: system acquisition; development; information system

1. Introduction

An information system is a framework of software, hardware, and procedures used to collect, store, manage, and process data into useful information for users. The main purpose of Information Systems is to provide fast and accurate access to information needed for decision making, operational management, and strategic analysis. Information systems can increase productivity, efficiency, and increase competitiveness that is increasingly competitive.

UIN Sunan Gunung Djati Journal House Information System is an important part of the university's academic and research programs. However, this system faces several problems that must be addressed immediately and effectively. The problem of inappropriate user interface (UI) and user experience (UX) is one of them. Due to the complex interface design and lack of attention to user comfort aspects, users often find it difficult to interact with the system. This hinders system adoption and reduces user efficiency in utilizing all available features.

In addition, there is a serious problem with sub-optimal integration between the components of Rumah Jurnal. Although the system has many features and modules, the lack of good integration can lead to data loss and hinder users from utilizing the full potential of the system. Therefore, efforts are needed to improve the integration between these components so that Rumah Jurnal can operate efficiently and effectively.

User Experience (UX) is very important because it can affect users' emotions, beliefs, behaviors, and achievements when using a product, system, or service. Many information technology companies in Indonesia implement a thorough user experience (UX) design process during the development cycle to improve the quality of the software they produce. Each company has different approaches and priorities for the UX design process, tailored to the specific needs in their product development [1].

© 2024 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 July 2024 doi:10.20944/preprints202407.1723.v1

In the context of user experience (UX), system integration plays a central role as it ensures that users can access information and use products or services with maximum continuity and efficiency. With good system integration, users can enjoy real-time data access, allowing them to get the latest information instantly. This not only supports more timely decision-making, but also improves user interaction with the product or service, creating a more seamless and satisfying experience. Effective integration also helps reduce bottlenecks in the usage process, allowing users to focus on their main objectives without being distracted by technical issues or lack of information [2].

Increasing system flexibility and adaptability to user needs is difficult due to the lack of space to change functionality in Rumah Jurnal. Users often have special needs that cannot be met by the existing system functionalities, and the existence of these limitations can prevent Rumah Jurnal users from fully using the system. Therefore, to make Rumah Jurnal an effective tool to support education and research at UIN Sunan Gunung Djati, continuous and planned improvements are needed.

2. Related Work

Research by Hossain (2023) describes an in-depth review of the various software development life cycle (sdlc) methodologies used in information systems (IS) projects. The goal is to provide an in-depth overview of the various software development life cycle (sdlc) methodologies used in the industry for information systems (IS) project management. There are several methods and algorithms discussed, namely the Waterfall Model, V-Model, Iterative Model, Agile Method, and Hybrid Model. The result of this research is to provide a comparative analysis between traditional SDLC methodologies (such as Waterfall) and Agile. Introduced the concept of a decision support matrix to help select the most suitable SDLC methodology based on factors such as project complexity, requirements stability, customer involvement, and the degree of flexibility required [3].

Research by Benvenuti et al. (2023) examined variations in the implementation of 21st century skills in basic education in different countries and the impact of new technologies such as VR, AR, gamification, and AI on education. The aim was to explore how these technologies can enhance creativity, critical thinking, problem solving and computational thinking among students and teachers. The methods used include an analysis of the literature regarding the application of technology in education as well as new reflections to help researchers, teachers, and educators understand the impact of technology on the development of human behavior and the acquisition of new skills [4].

This research develops a data acquisition and analysis platform for automated driving systems (ADS) based on cooperative perception of connected automated vehicles (CAVs). The goal is to process sensor data from multi-CAVs to extract the identity, position, velocity, and orientation of objects. The methods used include CAV sensor configuration, deep learning-based object detection algorithm with LiDAR, data fusion scheme, multi-object tracking method, Kalman filter, Chi-square test, and fuzzy logic approach to overcome non-continuous trajectory. Results show improved object detection and tracking as well as noise and outlier reduction, supporting traffic behavior analysis and ADS testing [5].

This research is about the development of an integrated system for regulating traffic in modern cities, with the aim of improving transportation efficiency, controlling energy-efficient LED street lights, and detecting and coordinating emergency vehicles. The goal is to create a single control unit that can manage, detect and report system faults, and improve transportation efficiency and emergency vehicle response. The methods used are the connection of all modules to a single control unit, the design of modules to control street lights and detect emergency vehicles, and the implementation of algorithms for coordination and reporting in traffic systems [6].

This research developed a systematic and quantitative framework for analyzing acquisition policies, aiming to address the heterogeneous needs of stakeholders and improve their understanding of the impact of acquisition policy options. The aim is to systematically identify stakeholders' needs, quantify the impact of policy alternatives, and provide a comprehensive analysis of various policy options. The methods used include needs identification, quantification process for evaluation, and

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 July 2024 doi:10.20944/preprints202407.1723.v1

holistic exploratory policy analysis. Case studies are included to illustrate the application of this framework in the introduction of a new military training system [7].

This study investigated the in-vitro diagnosis of cervical precancer using a handheld probe that has been automated with a LabVIEW-based control and acquisition system. The system is designed to detect polarization fluorescence and elastic scattering signals from the cervix. Aims to develop an automated system for more efficient and accurate diagnosis of cervical precancer using a handheld probe controlled by a motor to collect and analyze data in real-time. The method used is the use of motors to control the movement of the probe vertically, horizontally, and the change of polarization state. Data is collected through a LabVIEW-based GUI and analyzed in real-time using a MATLAB-based GUI [8].

This research develops a customer acquisition strategy for startups with a systematic approach that integrates inbound marketing principles, detailed buyer persona creation, customer journey mapping, lead generation process, and establishment of an efficient execution plan. The goal is to provide practical insights and recommendations to optimize customer acquisition, increase market visibility, and support sustainable growth in a competitive business environment. The method used involves the integration of inbound marketing principles, the creation of buyer personas, customer journey mapping, the implementation of a lead generation process, and the establishment of an execution plan, designed to provide structured guidance for companies to effectively attract and retain customers. [11]

3. Methodology

Methodology used in this reasearch is bespoke for acqustion and prototype for the development. Bespoke acquisition, in the context of this research, refers to the tailored and customized approach adopted for gathering the necessary data and resources. This methodology ensures that the acquisition process is specifically designed to meet the unique requirements of the project, rather than relying on generic or off-the-shelf solutions. The bespoke nature of this approach allows for a more precise alignment with the research objectives, enabling the collection of data that is highly relevant and context-specific. By customizing the acquisition process, the research team can account for variables and nuances that are unique to the study, thereby enhancing the accuracy and reliability of the results.

Furthermore, bespoke acquisition involves the integration of various techniques and tools that are selected based on their suitability for the specific research context. This could include the use of advanced technologies, specialized equipment, or unique data sources that are not commonly utilized in standard research methodologies. The tailored approach not only improves the efficiency of the data collection process but also ensures that the acquired data is of the highest quality. This, in turn, lays a robust foundation for the subsequent phases of the research, particularly the prototype development, by providing precise and comprehensive data that accurately reflects the research parameters.

Prototype development, as employed in this research, is a crucial phase that involves creating a preliminary version of the proposed solution or system. This phase is characterized by iterative design and testing processes, where the prototype is continuously refined based on feedback and performance evaluations. The primary objective of this phase is to transform theoretical concepts and design specifications into a tangible and functional model. This model serves as a proof of concept that demonstrates the feasibility and potential effectiveness of the proposed solution. Through iterative development, the research team can identify and address potential issues, thereby enhancing the overall robustness and reliability of the final product.

Prototype Software Development Life Cycle have step:

3.1. Requirement Analysis

At this stage, the developer thoroughly identifies and documents all the software and system requirements. This involves understanding the needs and expectations of the users and stakeholders,

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 July 2024 doi:10.20944/preprints202407.1723.v1

defining the functional and non-functional requirements, and setting the scope for the project. The goal is to ensure a clear understanding of what the system should achieve.

3.2. Creating a Prototype

In this phase, a temporary design of the system is created to demonstrate the flow and functionality from the user's perspective. This prototype is not a fully functional system but a visual and interactive representation to help visualize the final product. It focuses on the user interface and user experience, ensuring that the design meets the user's needs.

3.3. Prototype Evaluation

Evaluation is carried out to determine whether the prototype model meets the expectations and requirements of the users and stakeholders. Feedback is collected and analyzed to identify any discrepancies or areas for improvement. The aim is to refine the prototype until it aligns with the desired outcomes.

3.4. Coding the System

Once the prototype is approved, it is translated into the appropriate programming language. This involves writing the actual code to implement the system's functionality as defined in the requirements. Developers ensure that the code is efficient, maintainable, and adheres to best practices.

3.5. System Testing

After the software is coded, it must undergo rigorous testing to ensure it functions correctly. Various testing methods, such as White Box Testing, Black Box Testing, and others, are employed to identify and fix any bugs or issues. This step is crucial to ensure the reliability and performance of the system.

3.6. System Evaluation

The user evaluates whether the software meets the expected requirements and performs as intended. If the software meets the user's needs, the process moves to the next stage. If not, the system goes back to the coding and testing stages for further refinement based on the feedback received.

3.7. Using the System

Once the software has passed all evaluations and tests, it is ready for deployment and use. The final product is delivered to the users, who can start using it to achieve their intended tasks and objectives. Continuous support and maintenance may be provided to ensure smooth operation.

4. Result and Discussion

4.1. Result

4.1.1. Requirement Analysis

Requirement Analysis is the initial step in the Prototype Software Development Life Cycle. During this phase, stakeholders gather to discuss and define the system requirements. This includes understanding the needs and expectations of the end-users, identifying the functional and non-functional requirements, and setting the project scope. Detailed documentation is created to capture all the requirements, which will serve as a reference throughout the development process.

Stakeholders also prioritize the requirements to ensure that the most critical features are developed first. This prioritization helps in managing the project's timeline and resources effectively. Additionally, feasibility studies are conducted to assess the technical, financial, and operational viability of the project. This ensures that the project is practical and achievable within the given constraints.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 July 2024 doi:10.20944/preprints202407.1723.v1

Communication and collaboration are key during this phase. Regular meetings and discussions help in clarifying doubts, resolving conflicts, and ensuring that all stakeholders are on the same page. This collaborative approach helps in creating a comprehensive and clear requirement specification document, which serves as the foundation for the subsequent phases of the SDLC.

4.1.2. Creating a Prototype

Creating a Prototype is the next step where a working model of the system is developed. This prototype is a preliminary version of the software that demonstrates the core functionalities and user interface. It is built quickly and with minimal features to provide a tangible representation of the system.

The primary goal of the prototype is to gather feedback from the stakeholders, especially the end-users. By interacting with the prototype, users can better understand how the final system will work and provide valuable insights and suggestions for improvements. This iterative feedback loop helps in refining the requirements and making necessary adjustments early in the development process.

Prototyping also helps in identifying potential issues and challenges that might not have been apparent during the requirement analysis phase. By addressing these issues early on, the development team can avoid costly changes and delays later in the project. Overall, creating a prototype is an essential step that bridges the gap between the conceptual requirements and the actual system.

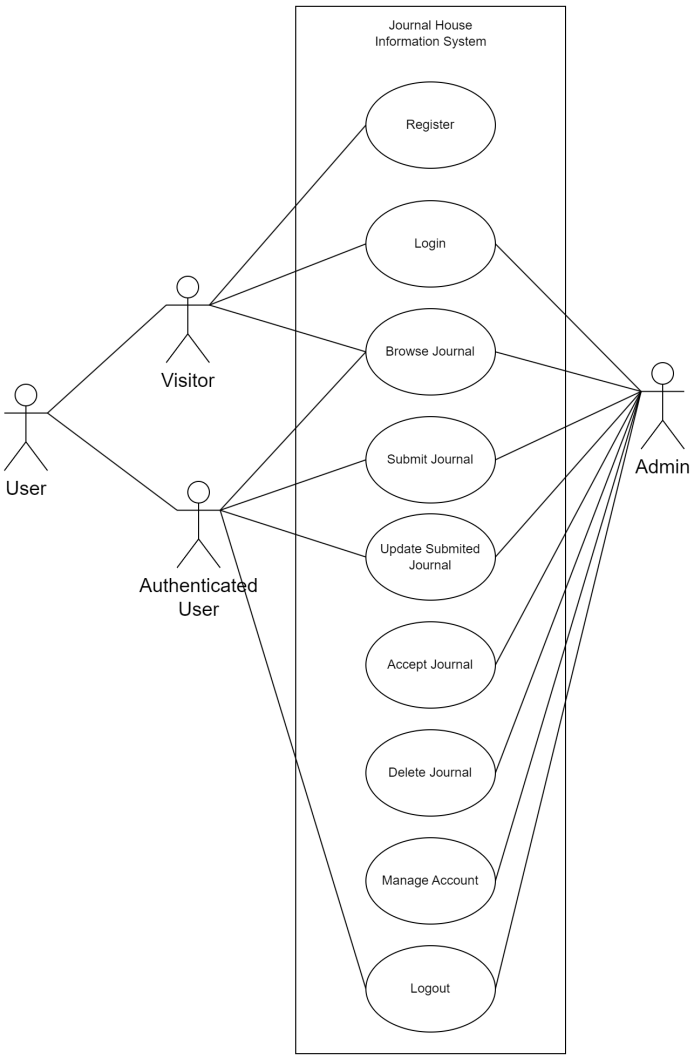

Figure 1. Prototype Use Case Diagram.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 July 2024 doi:10.20944/preprints202407.1723.v1

4.1.3. Prototype Evaluation

Prototype Evaluation involves presenting the prototype to the stakeholders for review and feedback. During this phase, the prototype is tested and evaluated to ensure it meets the initial requirements and expectations. Stakeholders, including end-users, project managers, and developers, interact with the prototype to identify any discrepancies, bugs, or areas for improvement.

Feedback gathered during this phase is critical for refining the prototype. This feedback helps in making informed decisions about changes and enhancements needed to better align the prototype with the users' needs. The evaluation process may involve multiple iterations, where the prototype is updated and reviewed several times to achieve the desired level of functionality and usability.

The evaluation phase also includes assessing the technical feasibility of the prototype. This involves testing the prototype's performance, scalability, and compatibility with existing systems. By thoroughly evaluating the prototype, the development team can ensure that the final system will be robust, user-friendly, and capable of meeting the project objectives.

4.1.4. Coding the System

Coding the System is the phase where the actual development of the software begins. Based on the refined prototype and the detailed requirements specification, developers start writing the code to build the system. This phase involves translating the design and functionality of the prototype into a fully functional software application.

Developers use various programming languages, tools, and frameworks to implement the system. They follow coding standards and best practices to ensure the code is efficient, maintainable, and scalable. This phase also includes integrating different modules and components to create a cohesive system that works seamlessly.

Collaboration and communication among team members are crucial during this phase. Regular code reviews, debugging sessions, and testing help in identifying and resolving issues promptly. By adhering to a structured development process, the team can ensure that the system is built to the required specifications and is ready for the next phase of testing.

4.1.5. System Testing

System Testing is a critical phase where the developed system undergoes rigorous testing to ensure it functions correctly and meets the specified requirements. Various types of testing, such as unit testing, integration testing, system testing, and user acceptance testing, are performed to identify and fix any defects or issues.

During unit testing, individual components or modules are tested in isolation to verify their functionality. Integration testing ensures that these modules work together as expected. System testing involves testing the entire system as a whole to check for overall performance, reliability, and compliance with the requirements.

User acceptance testing (UAT) is conducted to validate the system from the end-users' perspective. Users interact with the system to ensure it meets their needs and performs as expected in real-world scenarios. Any issues identified during testing are addressed and resolved to ensure the system is stable and ready for deployment.

4.1.6. System Evaluation

System Evaluation involves assessing the overall performance and effectiveness of the system after testing. This phase includes a thorough review of the system's functionality, usability, and compliance with the initial requirements. The development team evaluates the system to ensure it meets the project objectives and delivers the expected value to the stakeholders.

Feedback from the testing phase is analyzed to identify any remaining issues or areas for improvement. The team makes necessary adjustments and enhancements to optimize the system's performance

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 July 2024 doi:10.20944/preprints202407.1723.v1

and usability. This phase also includes reviewing the project management aspects, such as timelines, budget, and resource allocation, to ensure the project has been executed efficiently.

System evaluation is an ongoing process that continues even after the system is deployed. Continuous monitoring and feedback help in identifying opportunities for further enhancements and ensuring the system remains effective and relevant over time.

4.1.7. Using the System

Using the System is the final phase where the developed system is deployed and made available to the end-users. This phase involves transitioning the system from the development environment to the production environment. Deployment activities include setting up the necessary infrastructure, configuring the system, and migrating data from existing systems.

Once the system is deployed, users are trained to use it effectively. Training sessions, user manuals, and support documentation are provided to help users understand the system's features and functionalities. This ensures a smooth transition and minimizes any disruptions to the users' workflow.

Ongoing support and maintenance are essential to address any issues that arise after deployment. The development team provides technical support, bug fixes, and updates to ensure the system continues to function smoothly. By providing continuous support, the team ensures that the system remains reliable and effective, delivering long-term value to the stakeholders.

4.2. Discussion

The development of the prototype software system follows a structured and iterative approach that emphasizes continuous feedback and improvement. The initial phase of requirement analysis ensures that all stakeholders have a clear understanding of the project's scope and objectives. This collaborative approach helps in creating a comprehensive requirement specification document, which is crucial for the success of the project. By involving stakeholders early in the process, we can ensure that the final system meets their needs and expectations.

Creating a prototype serves as a critical step in bridging the gap between conceptual requirements and the actual system. The prototype provides a tangible representation of the system, allowing stakeholders to interact with it and provide valuable feedback. This iterative feedback loop helps in refining the requirements and making necessary adjustments early in the development process. As a result, potential issues and challenges can be identified and addressed promptly, avoiding costly changes and delays later in the project.

The evaluation of the prototype is essential for ensuring that the system meets the initial requirements and expectations. By involving stakeholders, including end-users, in the evaluation process, we can gather diverse perspectives and insights. This comprehensive evaluation helps in identifying discrepancies, bugs, and areas for improvement. Multiple iterations of the prototype ensure that it achieves the desired level of functionality and usability before moving on to the coding phase.

Coding the system involves translating the refined prototype and detailed requirements into a fully functional software application. This phase requires collaboration and communication among team members to ensure that the code is efficient, maintainable, and scalable. Regular code reviews, debugging sessions, and testing help in identifying and resolving issues promptly. By adhering to a structured development process, the team can ensure that the system is built to the required specifications and is ready for thorough testing.

System testing is a rigorous process that ensures the developed system functions correctly and meets the specified requirements. Various types of testing, such as unit testing, integration testing, system testing, and user acceptance testing, are performed to identify and fix defects or issues. This comprehensive testing approach helps in validating the system's performance, reliability, and compliance with the requirements. By addressing any issues identified during testing, the team can ensure that the system is stable and ready for deployment.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 July 2024 doi:10.20944/preprints202407.1723.v1

5. Conclusions

The prototype software development life cycle is a structured and iterative process that emphasizes continuous feedback and improvement. Involving stakeholders early in the development process ensures that their needs and expectations are understood and integrated into the system design. This iterative approach allows for ongoing refinement of requirements and early identification of potential issues. By soliciting feedback at each stage of prototyping and evaluation, development teams can address concerns promptly, resulting in a final system that is robust and user-friendly.

After thorough testing and evaluation, the finalized system is deployed for use by end-users. This deployment marks the culmination of a rigorous development process aimed at delivering a high-quality product. Ongoing support and maintenance are crucial to sustaining system reliability and effectiveness over time. Regular updates and enhancements based on user feedback and evolving technological trends ensure that the system continues to meet stakeholders' needs well into the future.

Following a structured development methodology and prioritizing continuous feedback are key to developing software systems that provide enduring value to stakeholders. By fostering collaboration between developers, stakeholders, and end-users throughout the software development life cycle, organizations can create solutions that not only meet immediate requirements but also adapt and grow with changing demands and technological advancements.

Acknowledgments: The author's wishes to express sincere gratitude to the Informatics Department of UIN Sunan Gunung Djati Bandung for their partial support and invaluable resources that facilitated this research work. Special thanks are extended to Mrs. Eva Nurlatifah, the course lecturer, whose guidance and expertise greatly contributed to the success of this paper. Furthermore, heartfelt appreciation goes to the dedicated team members who worked tirelessly and collaboratively to complete this research, as well as to all colleagues and friends whose encouragement and assistance were instrumental throughout the entire process.

References

1. I. Nugraha and A. Fatwanto, "User experience design practices in industry (case study from indonesian information technology companies)," *Elinvo (Electronics, Informatics, and Vocational Education)*, vol. 6, no. 1, p. 49–60, Sep. 2021. [Online]. Available: http://dx.doi.org/10.21831/elinvo.v6i1.40958
2. M. Rajabalinejad, L. van Dongen, and M. Ramtahalsing, "Systems integration theory and fundamentals," *Safety and Reliability*, vol. 39, no. 1, p. 83–113, Jan. 2020. [Online]. Available: http://dx.doi.org/10.1080/09617353.2020.1712918
3. M. I. H. , "Software development life cycle (sdlc) methodologies for information systems project management," *International Journal For Multidisciplinary Research*, vol. 5, no. 5, Sep. 2023. [Online]. Available: http://dx.doi.org/10.36948/ijfmr.2023.v05i05.6223
4. M. Benvenuti, A. Cangelosi, A. Weinberger, E. Mazzoni, M. Benassi, M. Barbaresi, and M. Orsoni, "Artificial intelligence and human behavioral development: A perspective on new skills and competences acquisition for the educational context," *Computers in Human Behavior*, vol. 148, p. 107903, Nov. 2023. [Online]. Available: http://dx.doi.org/10.1016/j.chb.2023.107903
5. X. Xia, Z. Meng, X. Han, H. Li, T. Tsukiji, R. Xu, Z. Zheng, and J. Ma, "An automated driving systems data acquisition and analytics platform," *Transportation Research Part C: Emerging Technologies*, vol. 151, p. 104120, Jun. 2023. [Online]. Available: http://dx.doi.org/10.1016/j.trc.2023.104120
6. S. Diwakaran, B. S. Kumar, K. Kumar, K. Sushmitha, and K. R. Chandra Siddhardha Reddy, "Design and development of multipurpose data acquisition for effective transportation," in *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*. IEEE, Mar. 2023. [Online]. Available: http://dx.doi.org/10.1109/ICACCS57279.2023.10112724

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 July 2024 doi:10.20944/preprints202407.1723.v1

7. K. Kwon, H. Kim, S. Min, C. Kim, C. Jee, and H. Seol, "Enhancing decision knowledge by developing a quantitative framework for holistic exploratory acquisition policy analysis," *Systems Engineering*, vol. 25, no. 5, p. 489–509, Jun. 2022. [Online]. Available: http://dx.doi.org/10.1002/sys.21632
8. A. N. Sah, S. Ahirwar, and A. Pradhan, "Developing an automated data acquisition system and real time analysis of an in house fabricated hand-held probe," in *Tissue Optics and Photonics*, Z. Zalevsky, V. V. Tuchin, and W. C. Blondel, Eds. SPIE, Apr. 2020. [Online]. Available: http://dx.doi.org/10.1117/12.2555719

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.