Jurnal Teknologi Informasi dan Pendidikan
Volume 18, No. 1, March 2025
https://doi.org/10.24036/jtip.v18i1.883

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

Design of Back-End Education Management System Application Using Iterative Incremental Method

Muhammad Hanan Rafif Farizta[1*]✉, Faishal Mufied Al Anshary[1], Syfa Nur Lathifah[1]
[1]Information System, Faculty of Industrial and System Engineering, Universitas Telkom, Indonesia
Corresponding Author:* ***hananfarizta@student.telkomuniversity.ac.id*

Article Information

Article history:

No. 883
Rec. July 31, 2024
Rev. February 02, 2025
Acc. February 04, 2025
Pub. March 17, 2025
Page. 664 – 677

Keywords:

- *Education Management System*
- *Education*
- *Back-End*
- *Iterative Incremental*
- *Load Testing*

ABSTRACT

Education plays a central role in shaping individuals and advancing society. One critical aspect of improving educational quality is the active involvement of parents, which significantly influences students' learning experiences at school. However, the main challenge lies in effective student data management and the lack of transparency in monitoring student progress. This research uses a case study at SMPN 1 Magetan and aims to design and implement the back-end of an Education Management System (EMS) application, focusing on the student module. By employing the Iterative Incremental method, this program is designed to be responsive to user needs and changes in the school environment. The Education Management System (EMS) is being developed to provide seamless access to student information. A scalable and efficient back-end is essential to prevent access difficulties as user demand grows. Load testing is conducted to ensure the system can handle concurrent requests reliably and maintain optimal performance. Development was carried out in two main phases, focusing on API usage efficiency. Rigorous testing, including load testing, ensured the application's readiness, with an average response time of 445 ms for 50 and 454 ms 100 simultaneous users. The Iterative Incremental approach proved effective in creating a responsive and adaptive educational application, supporting the development of a future generation of Indonesians competitive at the global level.

How to Cite:

Farizta, M. H. R., & et al. (2025). Design of Back-End Education Management System Application Using Iterative Incremental Method. Jurnal Teknologi Informasi Dan Pendidikan, 18(1), 664-677. https://doi.org/10.24036/jtip.v18i1.883

1. INTRODUCTION

Education is an intentional, well-thought-out attempt to completely mold people [1][2]. The educational process involves a number of elements, including teachers, learning environments, community support, and parental involvement. Education may also be described as the interaction between students and their surroundings that results in the transformation of captured information into knowledge [3]. Parents, students, and educational institutions can all gain from parental involvement in education, which can take many forms and be done both at home and at school [1][4].

Since education is essential to raising the caliber of human resources, it is always linked to development. While attempts have been made to raise the standard of education in developing nations such as Indonesia, the outcomes have not been ideal up to this point. Educational technology contributes to efforts to raise the standard of education in novel ways by concentrating on finding complex and related solutions to problems in the human learning process [5].

Technology has emerged as a key remedy in the modern era, demonstrating how technology is being progressively incorporated into Indonesian education. The intention is to equip the next generation, especially the students, with competitive skills that are comparable to those needed to meet global challenges so they can become innovators and carry on the nation's legacy, especially in the area of technology [6].

The research and analysis firm Gartner predicted that by 2020, roughly 60% of educational institutions would have completely switched to online-based systems. This suggests that, given the current circumstances, digital education is becoming more and more significant. With 4,500 higher education institutions and 165,000 elementary, middle, and high schools, Indonesia has enormous potential to create a cutting-edge, digitally based education system [7].

Digitalization has become a crucial aspect of transforming educational management systems from traditional paper-based methods to more efficient web-based systems. This shift is essential to ensure sustainable development within educational institutions [8]. Implementing web-based systems, such as student enrollment information systems, has been identified as an effective solution for enhancing educational management processes [9]. The process of digitalization in education involves converting traditional teaching methods into digital formats to achieve educational goals and objectives [10].

In light of these problems, this study will address how well features on the Education Management System (EMS) application's back-end are implemented, with a particular emphasis on features like attendance, permissions, dispensations, violation logbooks, complaint forms, and student profile. Designing the application back-end to be scalable over time, allowing for the inclusion of new features and increasing capacity while also ensuring data security through proper encryption and privacy protection [11]. The goal of this study is to determine how well load testing technique work when testing the EMS application back-end. Therefore, it is anticipated that the findings of this study will improve the educational environment at SMPN 1 Magetan and promote parental involvement in the learning process.

2. RESEARCH METHOD

A framework for carrying out the different stages of the research is provided by the research methodology [12]. The Iterative Incremental approach is utilized in this study article. An approach to system development called the Iterative Incremental method combines incremental and iterative methodologies. Iterative development is a process where software is developed repeatedly and improved upon with each iteration. By using an incremental method, software is created progressively in manageable chunks that can be independently evaluated and integrated. These two ideas are combined to create the iterative incremental technique in software development, which enables continuous development cycles. In these cycles, new features are added, bugs are fixed, and adjustments are made to the product as needed [13].

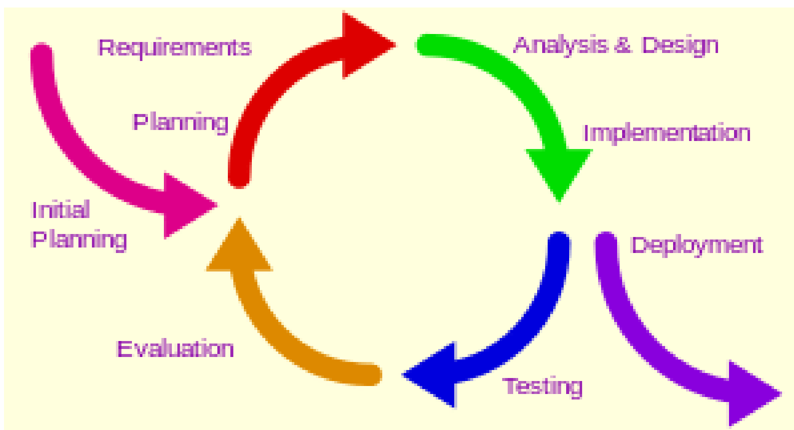


Figure 1. Iterative Incremental Phases

In Figure 1, Iterative incremental planning begins with requirements and initial planning. After that, a cycle of planning, analysis and design, implementation, testing, and evaluation will take place. Deployment will occur after this phase is finished. The Iterative Incremental technique is used to prioritize and sort requirements. This method focuses on completing the most important objectives at the beginning of the development process to produce a Minimum Viable Product (MVP), or initial product. This makes it possible for developers to get system user feedback earlier. Project managers must devote more time and resources to each new iteration. However, not all criteria can be anticipated or covered in the first analysis, which could lead to design challenges [14].

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

Jurnal Teknologi Informasi dan Pendidikan
Volume 18, No. 1, March 2025
https://doi.org/10.24036/jtip.v18i1.883

3. RESULTS AND DISCUSSION

3.1. Use Case Diagram

A use case diagram is a fundamental component in software engineering, serving as a requirements model that outlines how users (actors) interact with a system to achieve specified goals [15]. It is necessary to capture and describe the functional needs of a system [16]. In software development, use case diagrams are used to generate various UML artifacts, such as sequence diagrams and class diagrams, which aid in the visualization and understanding of system behavior and structure [17].

Figure 2. Use Case Diagram Implementation

Figure 2 depicts the design of the use case diagram, which tries to define the functions that users can execute depending on their previous requirements. The use case diagram is fully drawn, incorporating numerous aspects to be created.

Jurnal Teknologi Informasi dan Pendidikan
Volume 17, No. 1, March 2024
https://doi.org/10.24036/jtip.v18i1.883

3.2. Entity Relationship Diagram

Entity relationship diagram (ERD) are a fundamental tool in database design and conceptual modeling. They provide a visual representation of the relationships between entities in a database [18]. ERDs have been utilized in various domains, including software engineering, grounded theory inquiry, and manufacturing systems control [19][20]. The flexibility of ERDs allows for different interpretations and applications, such as in the automation of relational database normalization [21][22].

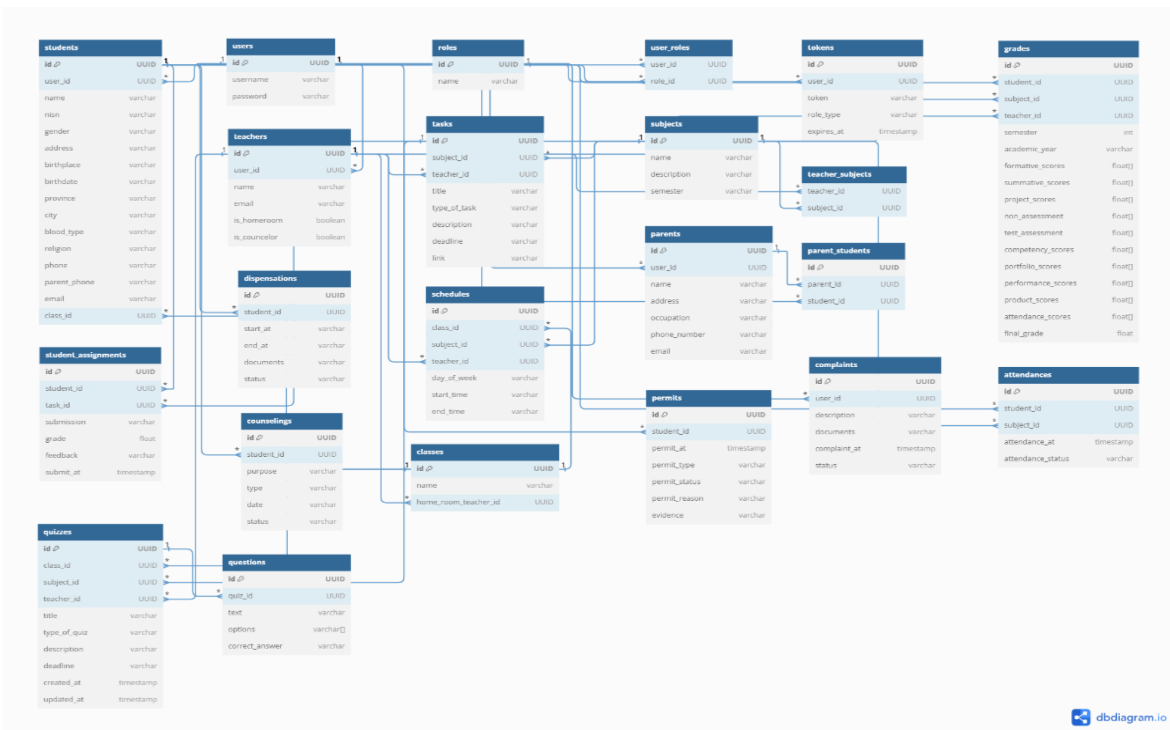


Figure 3. ERD Implementation

Figure 3 shows the design of the ERD for the EMS application. It includes 21 entities, namely: students, users, teachers, roles, user_roles, tokens, tasks, subjects, parents, dispensations, schedules, parent_students, grades, student_assignments, complaints, attendances, counselings, permits, classes, quizzes, and questions.

3.3. Iterative Incremental First Iteration

The first iteration of Iterative Incremental began on February 12, 2024, with the building of the Education Management System's back-end. During this stage, the back-end system was built utilizing the Go programming language, the Fiber framework, and

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

PostgreSQL as the database management system. The back-end system was developed throughout this period till March 17, 2024.

3.3.1. Planning

The planning stages are the initial phases in the design of features for the EMS application, particularly the student module features described in Table 1.

Table 1. First Iteration API Feature Plan

Code	Feature	Start Date	End Date	Status
UC001	View Attendance	Feb 12, 2024	Feb 18, 2024	TODO
UC002	View Schedule	Feb 19, 2024	Feb 25, 2024	TODO
UC003	View Profile	Feb 26, 2024	Mar 3, 2024	TODO
UC004	View Dispensation	Mar 4, 2024	Mar 10, 2024	TODO
UC005	View Violation Logbooks	Mar 11, 2024	Mar 17, 2024	TODO

3.3.2. Design

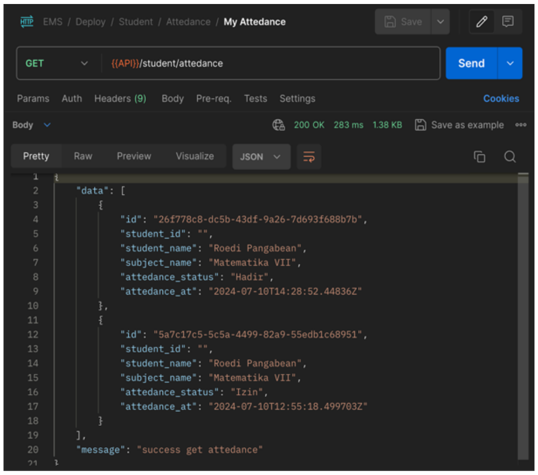
This phase is a crucial phase in the development of an information system, resulting in various important diagrams, such as use case diagrams and Entity Relationship Diagrams (ERD). These diagrams have been meticulously designed above and will serve as the primary foundation for developing the features of the EMS application.

3.3.3. Implementation

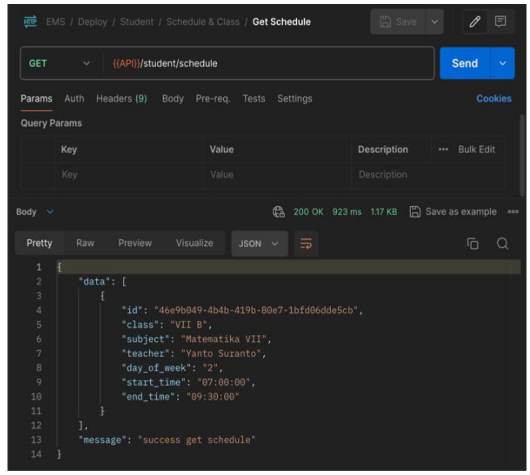
The implementation stage follows the planning and design stages in constructing the EMS application, with a focus on the student module elements outlined in Table 1 during the planning stage. Table 2 presents the final outcomes of the implementation stage.

Table 2. First Iteration of API Feature Implementation

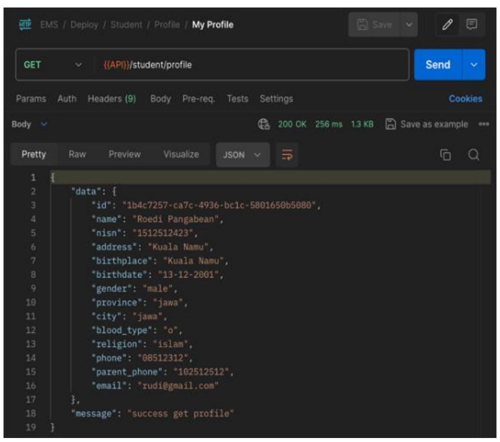
Code	Feature	Start Date	End Date	Status
UC001	View Attendance	Feb 12, 2024	Feb 18, 2024	DONE
UC002	View Schedule	Feb 19, 2024	Feb 25, 2024	DONE
UC003	View Profile	Feb 26, 2024	Mar 3, 2024	DONE
UC004	View Dispensation	Mar 4, 2024	Mar 10, 2024	DONE
UC005	View Violation Logbooks	Mar 11, 2024	Mar 17, 2024	DONE



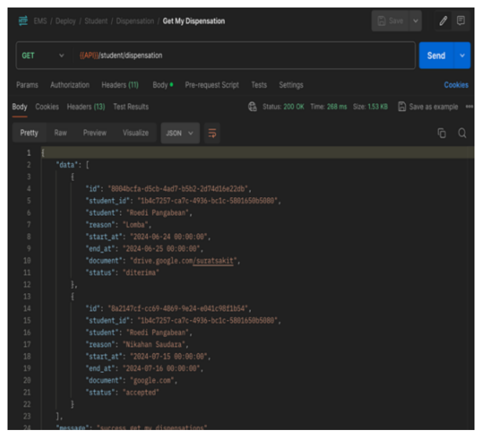
(a)



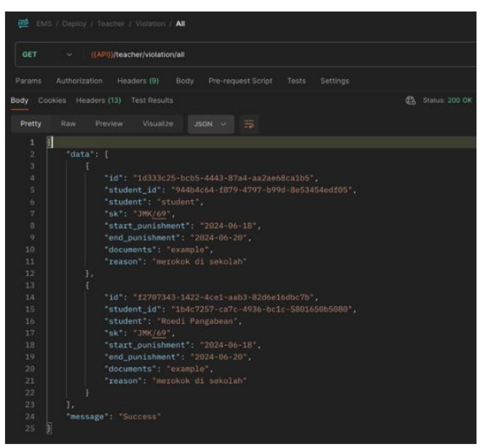
(b)



(c)



(d)



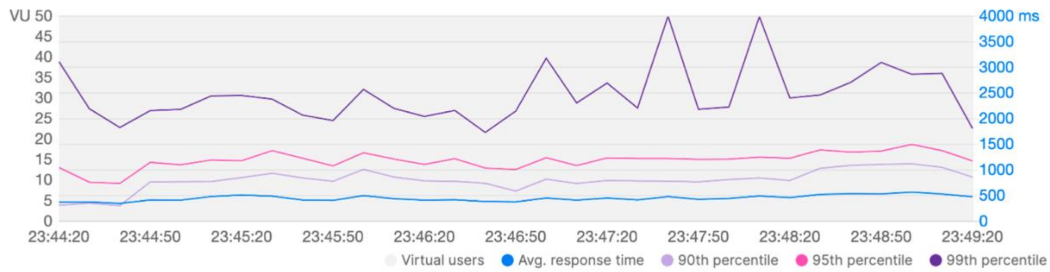
(e)

Figure 4. API Implementation for (a) view attendance, (b) view schedule, (c) view profile, (d) view dispensation, (e) view violation logbooks

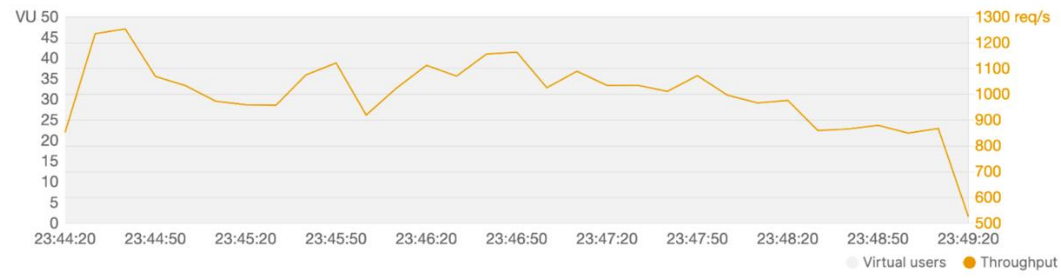
P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

3.3.4. Testing

In the first iteration, load testing was conducted to evaluate whether the code performance meets the requirements when accessed dynamically. This load testing involved 50 users simultaneously, each for a duration of 5 minutes.



(a)



(b)

Figure 5. Load Testing for 50 users resulted in (a) response time, (b) throughput

The load testing was conducted to evaluate the system's performance when accessed dynamically by 50 virtual users simultaneously for a duration of 5 minutes. The key performance metrics analyzed include response time, throughput, and error rate. Figure 5(a) illustrates the response time distribution across different percentiles. The average response time was recorded at 445 ms, while the 90th percentile response time remained relatively stable, indicating that most requests were handled efficiently. The 95th percentile response time showed slightly higher values, reflecting a small portion of slower responses, whereas the 99th percentile response time exhibited occasional spikes, suggesting performance fluctuations under peak load conditions.

Meanwhile, Figure 5(b) presents the throughput, which measures the number of requests processed per second. The system initially handled around 1200 requests per second, but as the test progressed, throughput showed a gradual decline, stabilizing around 900 requests per second before decreasing towards the end. Despite this fluctuation, the system maintained an average throughput of 101.48 requests per second throughout the test. Additionally, with 31,029 total requests processed and an error rate of 0%, the system demonstrated its ability to handle concurrent user requests effectively without failures. However, the spikes in response time at the 99th percentile suggest that some optimizations may be needed to ensure consistent performance, particularly under high load conditions.

3.3.5. Evaluation

Based on the test results, which were successfully completed, it was shown that all tests were concluded satisfactorily. In this second evaluation stage, the author concludes that the API in the Education Management System application operates as expected and is ready for use by the front-end.

3.4. Iterative Incremental Second Iteration

The second iteration of Iterative Incremental began on March 18, 2024, with the building of the Education Management System's backend. During this stage, the back-end system was built utilizing the Go programming language, the Fiber framework, and PostgreSQL as the database management system. The back-end system was developed throughout this period till April 14, 2024.

3.4.1. Planning

The planning stages are the initial phases in the design of features for the EMS application, particularly the student module features described in Table 3.

Table 3. Second Iteration API Feature Plan

Code	Feature	Start Date	End Date	Status
UC006	View Information	Mar 18, 2024	Mar 24, 2024	TODO
UC007	Submit Attendance	Mar 25, 2024	Mar 31, 2024	TODO
UC008	Submit Dispensation Approval	Apr 1, 2024	Apr 7, 2024	TODO
UC009	Submit Student Dispensation	Apr 8, 2024	Apr 14, 2024	TODO
UC010	View Student Permission	Mar 15, 2024	Apr 21, 2024	TODO

3.4.2. Design

This phase is a crucial phase in the development of an information system, resulting in various important diagrams, such as use case diagrams and Entity Relationship Diagrams (ERD). These diagrams have been meticulously designed above and will serve as the primary foundation for developing the features of the EMS application.

3.4.3. Implementation

The implementation stage follows the planning and design stages in constructing the EMS application, with a focus on the student module elements outlined in Table 3 during the planning stage. Table 4 shows the ultimate outcomes of the implementation stage.

Table 4. Second Iteration of API Feature Implementation

Code	Feature	Start Date	End Date	Status
UC006	View Information	Mar 18, 2024	Mar 24, 2024	DONE
UC007	Submit Attendance	Mar 25, 2024	Mar 31, 2024	DONE
UC008	Submit Dispensation Approval	Apr 1, 2024	Apr 7, 2024	DONE
UC009	Submit Student Dispensation	Apr 8, 2024	Apr 14, 2024	DONE
UC010	View Student Permission	Mar 15, 2024	Apr 21, 2024	DONE

(a) (b) (c) (d) (e)

Figure 6. API Implementation for (a) view information, (b) submit attendance, (c) submit dispensation approval, (d) submit student dispensation, (e) view student permission

Volume 17, No. 1, March 2024
https://doi.org/10.24036/jtip.v18i1.883

3.4.4. Testing

In the first iteration, load testing was conducted to evaluate whether the code performance meets the requirements when accessed dynamically. This load testing involved 100 users simultaneously, each for a duration of 5 minutes.

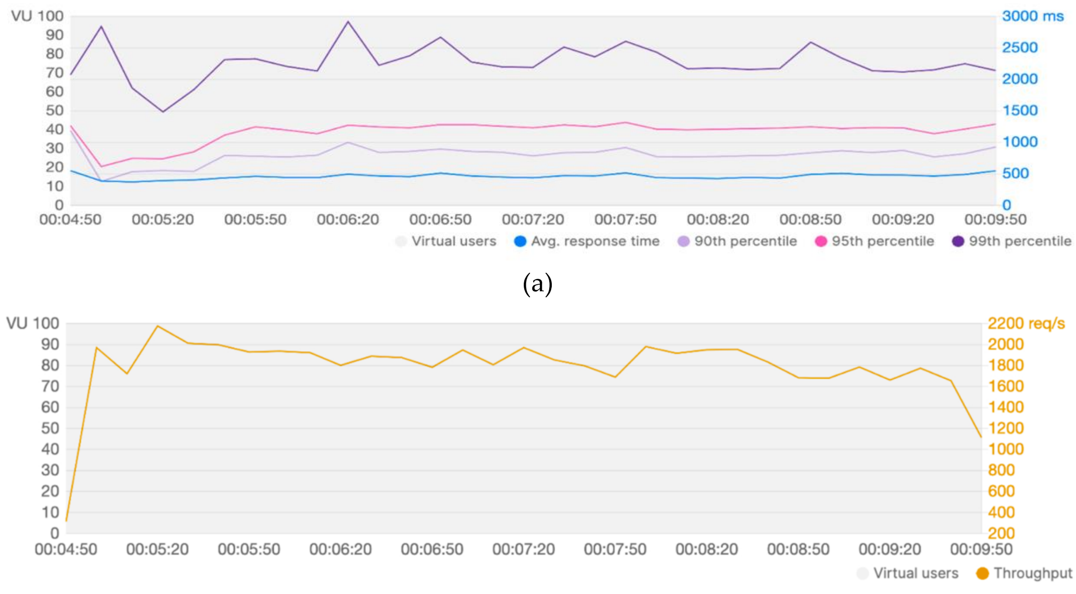


Figure 7. Load Testing for 100 users resulted in (a) response time, (b) throughput

The load testing was conducted to evaluate the system's performance when accessed dynamically by 100 virtual users simultaneously for a duration of 5 minutes. The key performance metrics analyzed include response time, throughput, and error rate. Figure 7(a) illustrates the response time distribution across different percentiles. The average response time was recorded at 454 ms, while the 90th percentile response time remained relatively stable, indicating that most requests were handled efficiently. The 95th percentile response time showed slightly higher values, reflecting a small portion of slower responses, whereas the 99th percentile response time exhibited occasional spikes, suggesting performance fluctuations under peak load conditions.

Meanwhile, Figure 7(b) presents the throughput, which measures the number of requests processed per second. The system initially handled around 180 requests per second, but as the test progressed, throughput remained relatively stable before decreasing towards the end. Despite this fluctuation, the system maintained an average throughput of 181.35 requests per second throughout the test. Additionally, with 55,394 total requests processed and an error rate of 0%, the system demonstrated its ability to handle concurrent user requests effectively without failures. However, the spikes in response time at the 99th

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

percentile suggest that some optimizations may be needed to ensure consistent performance, particularly under high-load conditions.

3.4.5. Evaluation

Based on the test results, which were successfully completed, it was shown that all tests were concluded satisfactorily. In this second evaluation stage, the author concludes that the API in the Education Management System application operates as expected and is ready for use by the front-end.

4. CONCLUSION

This research employs the Iterative Incremental method and uses the Go programming language to develop the back-end of the Education Management System application. There are two iterations that encompass features such as Attendance, Permissions, Dispensations, Violation Logbook, Complaint Form, and Student Personal Data. The application adopts an iterative and adaptive approach that responds to changing needs based on user feedback.

Load testing results indicate that all features have been successfully developed and meet the performance requirements. For 50 simultaneous users , the average response time was recorded at 445 ms, while for 100 simultaneous users, the average response time was 454 ms. Across both test scenarios, the system maintained a 0% error rate, demonstrating its ability to handle concurrent requests efficiently.

From these conclusions, it can be inferred that the Iterative Incremental approach is effective in the context of educational management application development, ensuring reliable performance and adaptability to user needs under varying load conditions.

REFERENCES

[1] S.-M. Bai and S.-M. Chen, "Evaluating students' learning achievement using fuzzy membership functions and fuzzy rules," *Expert Syst Appl*, vol. 34, no. 1, pp. 399–410, Jan. 2008, doi: 10.1016/j.eswa.2006.09.010.
[2] F. Harahap, N. E. A. Nasution, and B. Manurung, "The Effect of Blended Learning on Student's Learning Achievement and Science Process Skills in Plant Tissue Culture Course," *International Journal of Instruction*, vol. 12, no. 1, pp. 521–538, Jan. 2019, doi: 10.29333/iji.2019.12134a.
[3] E. Ramadansyah, R. Gelar Guntara, and A. Prehanto, "Design Thinking Approach for User Interface and User Experience on Campus Online Learning Platform," *Jurnal Teknologi Informasi dan Pendidikan*, vol. 17, no. 2, pp. 345–357, Sep. 2024, doi: 10.24036/jtip.v17i2.842.
[4] J.-H. Park, E. Lee, and S. H. Bae, "Factors Influencing Learning Achievement of Nursing Students in E-learning," *J Korean Acad Nurs*, vol. 40, no. 2, p. 182, 2010, doi: 10.4040/jkan.2010.40.2.182.

[5] D. S. Prawiradilaga, D. Ariani, and H. Handoko, *Mozaik Teknologi Pendidikan: E-Learning*, 1st ed., vol. 1. Jakarta: PrenadaMedia, 2016.
[6] A. R. Maulidi and A. Shalilah, "Pendidikan Anak Di Zaman Modern: Upaya Menghadapi Perkembangan Teknologi," *Jurnal Pendidikan Islam Anak Usia Dini*, vol. 1, no. 2, pp. 1–11, Dec. 2021, Accessed: Jul. 30, 2024. [Online]. Available: https://jurnal.stain-madina.ac.id/index.php/tila/article/view/537/513
[7] V. L. Ngongo, T. Hidayat, and Wiyanto, "Pendidikan Di Era Digital," *Jurnal Universitas PGRI Palembang*, pp. 628–638, Jul. 2019, Accessed: Jul. 30, 2024. [Online]. Available: https://jurnal.univpgri-palembang.ac.id/index.php/Prosidingpps/article/view/3093
[8] S. Kryshtanovych, O. Inozemtseva, O. Voloshyna, I. Ostapiovska, and O. Dubrova, "Modeling the Effective Digitalization of the Education Management System in the Context of Sustainable Development," *International Journal of Sustainable Development and Planning*, vol. 18, no. 5, pp. 1507–1514, May 2023, doi: 10.18280/ijsdp.180521.
[9] A. T. Widiyatmoko, A. Nugroho, and W. Wiyanto, "Development of Web-Based Student Registration Information System with Rapid Application Development Approach," *Journal of Computer Networks, Architecture and High Performance Computing*, vol. 6, no. 1, Feb. 2024, doi: 10.47709/cnahpc.v6i1.3459.
[10] A. N. Jibola Kadir and T. Abdulganiyu Adebayo, "Digitization in education system and management of Early Childhood Care Education in Nigeria," *Southeast Asia Early Childhood Journal*, vol. 8, no. 2, pp. 28–42, Nov. 2019, doi: 10.37134/saecj.vol8.no2.3.2019.
[11] W. Darwin, Z. Zulfadli, I. Yuliady, J. Jusmardi, and M. Deswina, "Designing Web-Based Mess and Dormitory Booking Applications," *Jurnal Teknologi Informasi dan Pendidikan*, vol. 17, no. 1, pp. 46–61, Dec. 2023, doi: 10.24036/jtip.v17i1.817.
[12] F. P. Kamarudin, "Optimizing Classification Algorithms Using Soft Voting: A Case Study on Soil Fertility Dataset," *Jurnal Teknologi Informasi dan Pendidikan*, vol. 16, no. 2, pp. 255–268, Dec. 2023, doi: 10.24036/jtip.v16i2.800.
[13] C. Larman and V. R. Basili, "Iterative and incremental developments. a brief history," *Computer (Long Beach Calif)*, vol. 36, no. 6, pp. 47–56, Jun. 2003, doi: 10.1109/MC.2003.1204375.
[14] A. E. Chowdhury, A. Bhowmik, H. Hasan, and M. S. Rahim, "Analysis of the Veracities of Industry Used Software Development Life Cycle Methodologies," *AIUB Journal of Science and Engineering (AJSE)*, vol. 16, no. 2, Jan. 2020, doi: 10.53799/ajse.v16i2.71.
[15] W. Shen and S. Liu, "Formalization, Testing and Execution of a Use Case Diagram," 2003, pp. 68–85. doi: 10.1007/978-3-540-39893-6_6.
[16] B. Anda, H. Dreiem, D. I. K. Sjøberg, and M. Jørgensen, "Estimating Software Development Effort Based on Use Cases — Experiences from Industry," 2001, pp. 487–502. doi: 10.1007/3-540-45441-1_35.
[17] T. P. Fries, "A framework for transforming structured analysis and design artifacts to UML," in *Proceedings of the 24th annual ACM international conference on Design of communication*, New York, NY, USA: ACM, Oct. 2006, pp. 105–112. doi: 10.1145/1166324.1166350.
[18] G. Della Penna, A. Di Marco, B. Intrigila, I. Melatti, and A. Pierantonio, "Interoperability mapping from XML schemas to ER diagrams," *Data Knowl Eng*, vol. 59, no. 1, pp. 166–188, Oct. 2006, doi: 10.1016/j.datak.2005.08.002.

[19] M. Brady and J. Loonam, “Exploring the use of entity-relationship diagramming as a technique to support grounded theory inquiry,” *Qualitative Research in Organizations and Management: An International Journal*, vol. 5, no. 3, pp. 224–237, Nov. 2010, doi: 10.1108/17465641011089854.

[20] K.-Y. Chen and S.-S. Lu, “A Petri-net and entity-relationship diagram based object-oriented design method for manufacturing systems control,” *Int J Comput Integr Manuf*, vol. 10, no. 1–4, pp. 17–28, Jan. 1997, doi: 10.1080/095119297131156.

[21] M. S. Patwardhan, P. S. Dhabe, A. A. Deshpande, S. G. Londhe, M. L. Dhore, and H. K. Abhyankar, “Diagrammatic Approach for Complete Automation of Relational Database Normalization at Conceptual Level,” *International Journal of Database Management Systems*, vol. 2, no. 4, pp. 132–151, Nov. 2010, doi: 10.5121/ijdms.2010.2411.

[22] D. P. S., P. M. S, D. Asavari A., D. M. L, B. B.V, and A. H. K., “Articulated Entity Relationship (AER) Diagram For Complete Automation Of Relational Database Normalization,” *International Journal of Database Management Systems*, vol. 2, no. 2, pp. 84–100, May 2010, doi: 10.5121/ijdms.2010.2206.