

Rubrick Marking Application (RMA): Development of Hybrid Database Solution for Structured and Unstructured Data

Mohd. Faiz Tony^{1*}, Athirah Musa¹, Nurzaihan Ismawi¹

¹Department of Information and Communication Technology, Politeknik Negeri Mukah,
KM7.5 Jalan Oya, 96400 Mukah, Sarawak, Malaysia

*Corresponding author: mohdfaiztony@gmail.com

Abstract

The Rubrick Marking Application (RMA) addresses critical inefficiencies and biases inherent in traditional educational assessment methods. This web-based tool was developed to streamline and enhance the assessment process, specifically targeting challenges identified within educational settings. This study provides a comprehensive overview of the design, development, and testing phases of the RMA. Validation through User Acceptance Testing (UAT) involved 26 administrators and assessors, confirming the RMA's alignment with user requirements and its significant facilitation of the marking process, particularly for large groups. The RMA's design emphasizes transparency and minimizes bias, ensuring consistent and fair grading practices. A hybrid database storage approach utilizes MariaDB's capabilities to store unstructured data in JSON objects, combining relational database robustness with NoSQL flexibility to accommodate diverse rubric formats. Moreover, the RMA's core many-to-many relationship module enhances scalability and adaptability, extending its applicability beyond educational contexts to various assessment needs. As educational institutions increasingly adopt digital solutions, the RMA represents a substantial advancement in educational technology. By addressing inherent inefficiencies and biases, it provides a robust, transparent, and scalable tool to significantly enhance the assessment process. This research underscores the necessity and rationale behind the development of the RMA in response to identified challenges within educational assessment methodologies.

Keywords: - marking application, hybrid database, rubric, nosql, structured data, unstructured data

1. Introduction

The incorporation of digital technology in educational assessments has greatly altered traditional grading methods, improving accuracy and fairness in assessing student achievements. This article explores the development of a rubric-based grading tool that incorporates SQL which is highly efficient for transactions requiring high integrity and complex querying capabilities (Elmasri & Navathe, 2015) and NoSQL databases which is used to manage unstructured data, offering scalability and flexibility (Gandomi & Haider, 2015). The creation of this hybrid database system aims to address the evolving needs of modern educational environments. It enables adaptable and efficient data management to facilitate the overall marking process.

Rubric-based evaluations employ predefined criteria to ensure consistent and unbiased assessment of student tasks (Brookhart, 2013). Traditionally, these evaluations have been carried out using paper-based techniques or simple digital devices, which frequently face challenges related to scalability, data handling, and immediate processing (Cattell, 2011). Dealing with these difficulties requires a fresh method of managing databases that can efficiently and flexibly handle large amounts of data (Stonebraker & Çetintemel, 2005). The suggested application uses a

hybrid database system that merges the advantages of SQL and NoSQL databases, utilizing SQL for its strong transactional support and structured query abilities, and NoSQL for its scalability and adaptable schema design (Han et al., 2011).

SQL databases are renowned for their relational model and ACID (Atomicity, Consistency, Isolation, Durability) properties, which provide a stable and reliable foundation for managing structured data and ensuring data integrity (Elmasri & Navathe, 2015). They are particularly effective in scenarios requiring complex queries and transactional operations. However, their rigid schema design can become a limitation when dealing with unstructured or semi-structured data, increasingly prevalent in educational assessments (Stonebraker, 2010).

In contrast, NoSQL databases, such as MongoDB and Cassandra, offer flexible schema designs and horizontal scalability, making them suitable for handling large-scale data and dynamic data models (Sadalage & Fowler, 2012). These features are particularly beneficial for storing diverse types of assessment data, including multimedia files, annotations, and varied rubric formats. The ability to scale out by adding more nodes to a NoSQL database also addresses the performance bottlenecks often encountered in traditional SQL databases (Han et al., 2011).

By integrating SQL and NoSQL databases, the rubric marking application aims to optimize performance and flexibility. The SQL component ensures reliable data storage and complex query processing, while the NoSQL component accommodates the evolving data structures and high-speed access required for real-time assessment. This hybrid approach not only enhances indexing speed but also maintains the adaptability necessary for diverse educational contexts (Moniruzzaman & Hossain, 2013; Cattell, 2011).

The development of this hybrid database system underscores the importance of adopting versatile technological solutions in educational settings. The architecture of the rubric marking application is designed to support scalability, reliability, and efficient data retrieval, addressing the specific needs of educational institutions (Santos et al., 2015; Sharma et al., 2015). This paper provides a comprehensive examination of the architectural design, implementation strategies, and offering insights into its potential impact.

2. Literature Review

Recent advancements in database technologies have significantly facilitated the integration of structured and unstructured data. This section examines contemporary case studies demonstrating the advantages of integrating both data types in educational assessment, marking, and grading.

Smith et al. (2020) explored the implementation of an integrated assessment platform at a leading educational institution. This platform seamlessly combines structured data, such as student records and exam scores, with unstructured data, including essay submissions and multimedia assignments. By integrating these data types, educators gain comprehensive insights into student performance, enabling personalized feedback and targeted intervention strategies (Smith et al., 2020). The scalability of NoSQL databases for managing unstructured data ensures that the platform can accommodate the increasing volume and variety of assessment materials generated by students (Smith et al., 2020).

Chen et al. (2021) investigated the development of an automated grading system in a secondary school setting. Leveraging machine learning algorithms, the system analyzes structured data from multiple-choice tests and short-answer questions, significantly reducing grading time and improving accuracy (Chen et al., 2021). By automating the evaluation process for objective questions, teachers can allocate more time to providing qualitative feedback on subjective responses (Chen et al., 2021). The system's ability to analyze large datasets of student responses enhances grading consistency and minimizes subjective biases (Chen et al., 2021).

García et al. (2019) examined the deployment of a learning analytics platform in a university environment. This platform utilizes unstructured data from student forums, social media interactions, and course materials to predict student performance and engagement (García et al., 2019). Using natural language processing techniques, the platform accurately predicts student outcomes, allowing educators to intervene proactively and provide targeted support (García et al., 2019). Insights derived from unstructured data enable personalized learning experiences tailored to individual student needs and preferences (García et al., 2019).

While these case studies illustrate the diverse applications of structured and unstructured data in educational assessment tools, they lack sufficient discussion on the role and importance of rubrics. Rubrics are essential for effective assessment practices as they provide a clear framework for evaluating student performance, ensuring transparency and consistency (Andrade, 2005). By offering specific criteria and detailed performance levels, rubrics help students understand expectations and improve their work (Reddy & Andrade, 2010). Incorporating rubrics into assessment platforms can enhance the reliability of evaluations by providing detailed criteria against which student work is measured (Brookhart, 2013). Moreover, rubrics facilitate self-assessment and peer feedback, encouraging students to engage more deeply with the learning process (Panadero & Jonsson, 2013).

The inclusion of rubrics in automated grading systems and learning analytics platforms can further refine the assessment process. For instance, rubrics can be integrated into machine learning algorithms to evaluate more complex student outputs, such as essays or projects, thereby combining the efficiency of automated systems with the nuanced understanding provided by human evaluators (Stevens & Levi, 2013). This integration ensures that the assessment tools are not only scalable and efficient but also maintain high standards of fairness and educational value.

Emphasizing the significance of rubrics would enrich the discussion and provide a more comprehensive understanding of assessment methodologies. Rubrics serve as a bridge between the qualitative aspects of student work and the quantitative metrics used in educational technologies, ensuring that assessments are holistic and educationally meaningful.

3. Methodology

The methodology can be defined as the systematic, theoretical analysis of the method implemented to design, develop, and maintain a system. The implementation of the correct methodology can ensure the system development is on track and can be done gradually, effectively, and systematically. It

should be noted that there are many types of methodology that can be used in system development.

In this study, Rapid Application Development (RAD) methodology is implemented to reduce development time through iterative prototyping and frequent user feedback, which is particularly beneficial for small-scale applications where quick delivery is essential. RAD's flexibility allows developers to adapt to new requirements seamlessly, which is vital for small-scale projects that often face changing user needs and market conditions. This adaptability ensures that the final product remains relevant and aligned with user expectations (Sommerville, 2020). The RAD methodology consists of 4 phases which are requirement planning phase, user design phase, construction phase and cutover phase as shown in Figure 1.



Fig. 1. Rapid Application Development Model

3.1 Requirement Planning Phase

In this phase, the initial requirement gathering process was conducted to construct the conceptual framework of the application. This phase was crucial as the developer needed to construct the expectation of the client from an imaginary concept to actual needs. This phase determined the backbone of the application and the technologies that will be implemented to suit the application operation. Various well-established applications reviewed with the client to narrow down the scope and specification alongside the technological implementation. All the gathered information in this process will be used in the next phase which consists of multiple iteration of prototyping.

3.2 User Design Phase

The user design phase consists of an iteration of multiple processes which are prototyping, testing, and refining as shown in Figure 2. In this process the developer is tightly coupled with the client on the outcome of the iteration. Start from the first iteration until the last iteration of the process, the development of the application will be refined to accordingly following the initial specification set in the previous phase. The focus of this phase was to develop a working prototype of the application module by module based on the operation of the application whether it meets the client's need or not. For a small-scale application, the testing process was conducted parallel with the development to achieve a faster production process. A faster development time contributes to the enhancement of the application in

which the refinement process will involve the client operating it. The iteration of the process is complete once every module has been developed, tested, and refined accordingly.

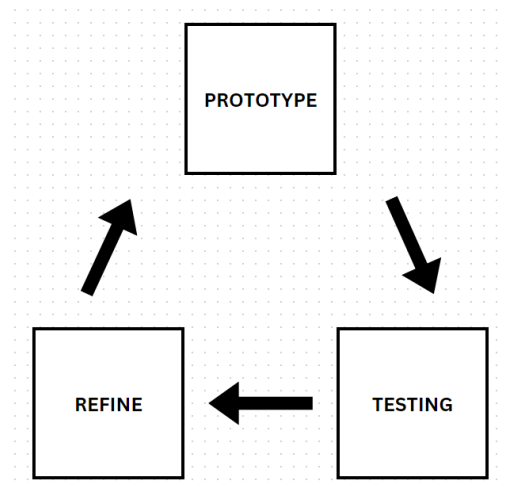


Fig. 2. User Design Iteration Process

3.3 Construction Phase

This phase includes final construction of the user interface of the RMA. In the iteration of user design process, it focusses heavily on the input and output of the function or module to ensure the RMA work as intended meanwhile in this phase, the product of the user design process is further refined by constraining the user interface to cater the requirements. The final user interface was produced in this stage and the database design of the RMA was also redefined to accommodate the technological implementation.

3.4 Cutover Phase

This phase involves the moving of the application to the production environment, which involves setting up the necessary infrastructure, configuring the system, and ensuring all components are properly integrated (Sommerville, 2020). All the integration of the module developed, and the technology implemented is tested in the production environment. This phase will include the process of user training, final documentation, and deployment to the production server.

4. System Design and Development

4.1 Core Database Design

The data storage design of the RMA is composed of 4 separate building blocks which are assessee block, assessor block, rubric block, and group block. The operation of the RMA relies on the many-to-many relationship of each module as shown in Fig. 3.

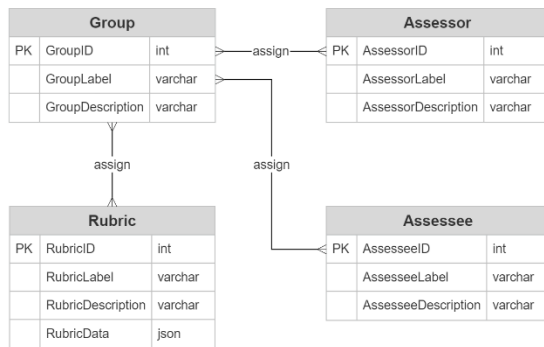


Fig. 3. Entity Relationship Diagram of Database

The main database engine chosen for the RMA based on the specification is the relational database management system or RDBMS which is commonly referred to the SQL. This structured data storage is used on all modules. The implementation of the RDBMS to store most of data was due to its ACID (Atomicity, Consistency, Isolation, Durability) compliance, which is crucial for applications where data consistency and transaction reliability are paramount. Additionally, RDBMSs utilize structured schemas, which facilitate complex joins and queries, making them ideal for relational data and applications that require robust transactional support (Taipalus, 2024).

Although there is no NoSQL database management engine such as MongoDB involved in this development but the idea of using the hybrid approach which is using structured and unstructured data storage is possible for the current generation of RDBMS. This technique utilizes the JSON (JavaScript Object Notation) field type of SQL in storing the unstructured data. It is a common data format used in NoSQL databases is JSON (JavaScript Object Notation). JSON is a lightweight, text-based format that is easy for humans to read and write, and easy for machines to parse and generate. This format is particularly well-suited for NoSQL databases, like MongoDB, because it supports nested structures and allows for the flexible and dynamic storage of data. JSON's flexibility enables the storage of complex data types and structures without the need for predefined schemas, making it ideal for applications with evolving data models (Taipalus, 2024).

4.2 Use Case Design

Use case design plays a critical role in software development by capturing functional requirements and detailing interactions between users (actors) and the system. This process aids in identifying and addressing potential issues early, ensuring alignment between business objectives and technical requirements. Use cases help visualize system interactions, manage project scope, and facilitate

communication among stakeholders, leading to better project planning and risk management.

4.2.1 Level 0 Use Case

In the foundational stage, Level 0 involves the exhaustive identification of all actors within a system alongside their hierarchical relationships, thereby delineating the involved participants (Kulak & Guiney, 2016). Figure 4 below displays the top-level use case diagram of the RMA.

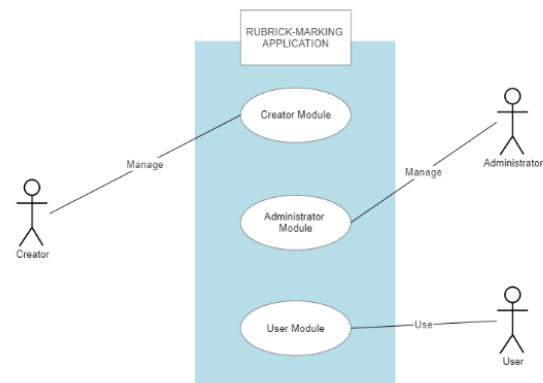


Fig. 4. Level 0 Use Case Diagram of RMA

Based on Fig. 3, the RMA consists of 3 main users which are the creator, administrator, and user. Each of the actors has its distinct goals in which the creator actor goals are associated with the creator module, administrator actor is associated with the administrator module and the user module is associated with the user module.

4.2.2 Level 1 Use Case

Subsequently, at Level 1, the use case model encapsulates the most overarching use cases, serving as a depiction of the primary business processes facilitated by the software system in development (A. Cockburn & L. Cockburn, 2008). Each use case represents a distinct business process and serves as a foundational element in system design and development.

4.2.2.1 Creator Module Use Case

The use case design shown in the Fig. 5 illustrates the goals of the creator in the RMA. Based on the use case diagram, the creator which is basically the owner of the RMA wild held responsibility in the creation of the administrator. This includes several functionalities such as registration process, update process and the delete process. One of the examples of this goals is to assign an administrator of the RMA for an institution, so any usage of the RMA can be handles by the administrator. This application does not limit to single

scenario only but applicable for several other scenario where a rubric based marking tools is needed such as a pitching competition, innovation competition and so on.

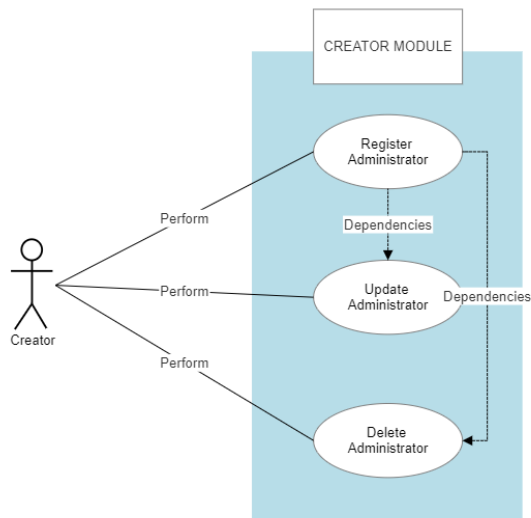


Fig. 5. Level 1 Use Case Diagram of Creator

4.2.2.2 Administrator Module Use Case

The major application operation lies in the administrator module as shown in Figure 6 in which there are multiple goals associated with the actor and the dependencies in achieving the goals.

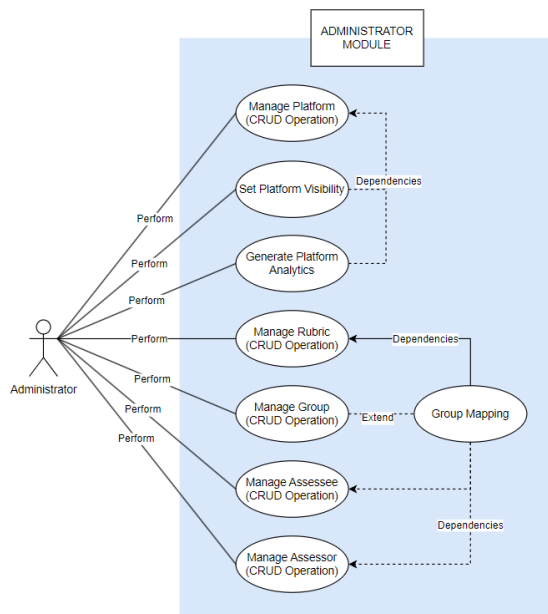


Fig. 6. Level 1 Use Case Diagram of Administrator

Based on the diagram, the CRUD operation stated in the goals reflect to the database operation of the associated goals common data. The CRUD stands for Create, Read, Update and Delete which is the common

functionality of any information system. For example, for the Manage Platform goal, the CRUD operation will include the creation of platform, displaying the platform, updating the platform, and deleting the platform.

In the diagram, the term platform refers to the marking platform. This can be any activities such as a competition, assessment, or interview which will used the RMA as the assessment tool. Meanwhile the rubric is referred to the marking rubric that will be used in this platform. The marking rubric is not limited to a single rubric and due to the many-to-many relationship of the database design, multiple rubric able to be utilize. Since the database of the rubric implementing the structured and unstructured data storage, the common data for the rubric such as rubric name and its description is stored under the structured data section meanwhile the rest of the rubric data such as weightage, criteria, criteria content, label, and marks are stored in the unstructured data section.

The main reason why we implemented this hybrid data storage was due to the nature of the rubric itself. The rubric that is dynamically changes with no fixed weightage, number of criteria, the marks and other data required a high flexibility of data storage offered in the NoSQL database. This implementation reduces the risk of making the database too complex and maintaining the simplicity of the database. In general, all the data that can be indexed is stored in the structured database meanwhile the mixed data is stored in JSON form for further application processing.

Next is the group, the initial scenario of the RMA when it is developed is it able to cater a multiple group that contain multiple different type of jury or assessor and multiple project title or assessee. Thus, the function of the group was to group assessor and assessee in a category or group depending to the platform configuration. This process is the mapping process which is the main feature of the RMA.

On other hand, the mapping process of the group dependent on the assessee, assessor, and the rubric. Each group created by the administrator in the platform must be assigned with one or many rubrics, assessee, and assessor.

4.2.2.2 User Module Use Case

The user is the end-user of the RMA which refer to assessee or jury who will directly interact with the application to process the marking. The following Fig. 7 show the use case diagram of the user actor and its goals. Based on Fig 7, the display assigned group refer to the group that has been assigned to the user in the previous administrator section. Only the assigned group is visible to the user as this feature ensure the integrity of the data. In each of the group, the user will

be able to see the assigned assessee list alongside with its status either marked or not. As for the create marking and update marking, the RMA will fetch the assigned rubric and displayed it to the user for the user to select the marks.

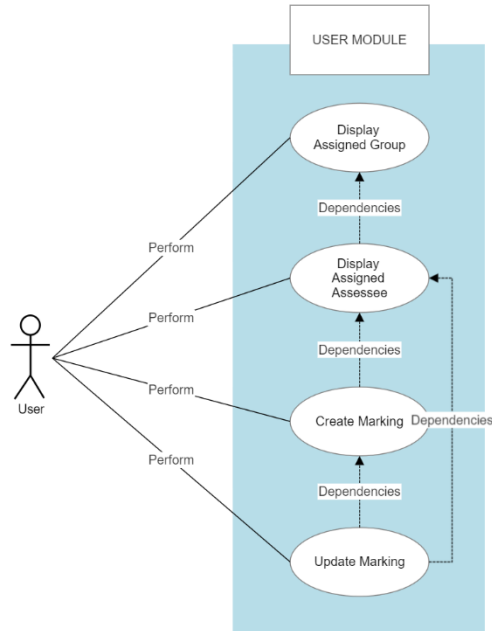


Fig. 7 Level 1 Use Case Diagram of User

The group that is visible to the user does not limit to a single group. It can be a multiple group and the rubric for each group can be different depending on the mapping made by the administrator.

5. System Implementation and Testing

The front-end section of the RMA was developed using the Bootstrap 5 framework alongside with the jQuery to handle the dynamic user interaction with the marking rubric. Meanwhile on the back-end part, RMA is utilizing the Codeigniter PHP framework as the main framework due to its lightweight properties. As for the database, MariaDB was used as the main database engine and as for the version used which is the 10.6.8 it offered a JSON datatype which is fully utilized to stored unstructured data.

5.1 RMA Graphical User Interface

This section will primarily focus on the mapping process, rubric creation process and marking process of the RMA which involved the administrator and the user.

5.1.1 Administrator User Interface

Fig. 8 shows the mapping process of the group. each group will need to be assign with rubric, assessor, and

assessee to be fully qualified to be used by the user. Each of the component are dependent on the available data on the platform since each platform has its own rubric, assessor and assessee which need to be complete before the group creation is made.

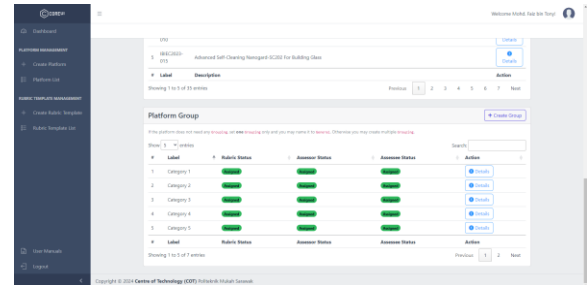


Fig. 8 Group Mapping User Interface

As all the process complete the analytic section will be visible to the administrator, enable the administrator to gather the insight of the group as shown in Fig. 9. The analytic section is available for each group, and it is not visible to the assessor although they are the one who creating the marking. This feature is developed intentionally to prevent marks moderation by the assessor. Although the marks moderation is not fully prevented but the way the system is designed will require a lot of work in assessor side to review each mark.

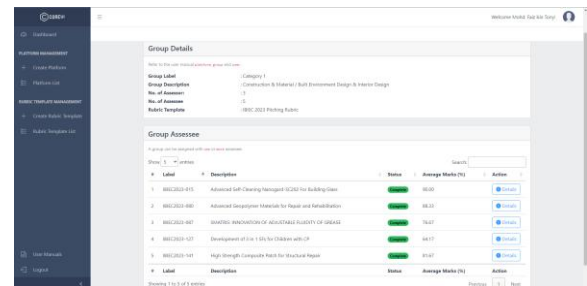


Fig. 9 Group Analytics User Interface

Moving on to the next graphical user interface display in Fig. 10. It displays the user interface of the rubric creation process. the rubric template details are data that is stored in the structured database section meanwhile the rubric template configuration is the part that is stored in the unstructured database part in form of JSON object. In the interface shown, the administrator is free to modify the rubric dynamically and it is possible to store the data in the normal SQL database, but the complexity of the data will take up a lot of resources thus JSON is chosen to cater the rubric data. This will provide application extensibility in the development of the project since JSON is commonly used in the industry and a lot of platform support JSON object.

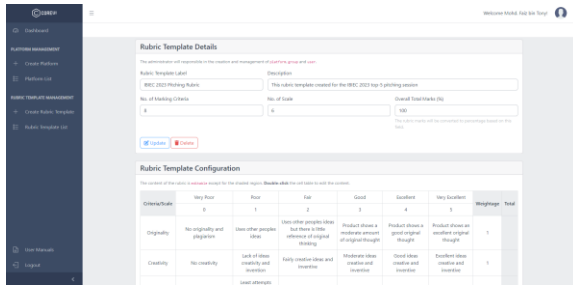


Fig. 10. Rubric Creation User Interface

5.1.2 Assessor User Interface

The design idea of the assessor user interface was to make it as simple as possible as one of the requirements the client needed was to cater a non-IT literate person. In achieving this goal, the user interface of the assessor is made up components that are needed for marking process only as shown in Fig. 11.

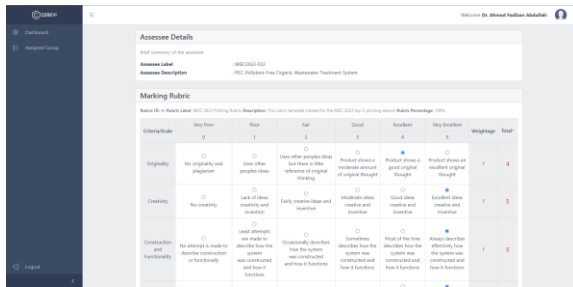


Fig. 11. Marking Process User Interface

The rubric that is been display for an assessee in which the only available action or function for the assessor was to submit the mark and to set the mark only. This design implementation highly reiterated in the user design process to cater the non-IT person. This simplicity also has been programmed to minimize the risk of human error. A lot of data validation process conducted for the marks submission to maintain the integrity of the data in the JSON object.

5.2 RMA User Acceptance Test (UAT)

After the system is fully developed, designed, and deployed in web hosting, User Acceptance Test (UAT) is conducted. According to the previous studies by Afrianto et al. (2021) and Sandin et al. (2016), UAT is typically performed at the final stage of development to ensure that the system can effectively handle required tasks in real-world scenarios. Martillano et al. (2018) also emphasize the importance of UAT in system development, advocating for involving

between five to ten related users to validate system functionalities.

In our implementation of the RMA in a real scenario, 26 users comprising assessors and administrators of the RMA were selected to conduct the UAT. The objective was to comprehensively test all RMA functionalities and ensure that the system performs as expected under operational conditions.

Upon completion of the UAT, the gathered results were analysed to ascertain user expectations towards RMA functionalities. The results, as depicted in Figure 12, indicate that the RMA successfully passed the UAT phase. Feedback from all users confirmed a clear understanding of how the RMA operates and how to utilize its functionalities. Furthermore, extensive testing of all system features revealed no logic errors, demonstrating robust functionality.

Moreover, users expressed satisfaction with how the RMA facilitates the marking process, providing transparency throughout the application lifecycle. This positive reception underscores the system's effectiveness in meeting user needs and expectations in a real-world setting.

To further elaborate on the UAT findings, it is crucial to emphasize that user acceptance testing not only validates technical performance but also assesses user experience and satisfaction. By involving a diverse group of stakeholders in the testing phase, we ensured that the RMA aligns with user requirements and enhances operational efficiency.

In conclusion, the successful UAT results validate the RMA's readiness for deployment, affirming its functionality and user-friendliness. Future enhancements could focus on refining user interfaces based on feedback gathered during testing, thereby continually improving user satisfaction and system performance.

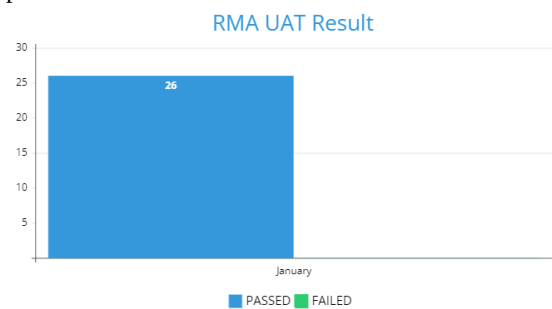


Fig. 12. RMA UAT Result

6. Discussion and Conclusion

The Rubrick Marking Application (RMA) has been designed, developed, and thoroughly tested to enhance the marking and assessment process in educational settings. This web-based application underwent extensive User Acceptance Testing (UAT) with a group of 26 targeted users, comprising both administrators and assessors. The UAT aimed to validate all functionalities of the RMA, ensuring it meets the requirements and expectations of its users. The results depicted in Fig. 12, indicate that the RMA is fully functional and satisfies user requirements effectively.

Users reported that the RMA significantly simplifies the marking process, particularly when managing assessments for large groups of students. Traditional paper-based or less structured digital marking systems often become cumbersome and error-prone as the number of assessee increases. In contrast, the RMA provides a streamlined, efficient process that reduces the administrative burden on educators. Moreover, the application's intuitive interface and clear navigation facilitate ease of use, even for those with limited technical skills.

One of the critical advantages of the RMA is its ability to enhance transparency in the marking process. By digitizing the rubric and assessment criteria, the RMA ensures that all assessors adhere to the same standards, thereby minimizing the potential for subjective bias. This uniformity is particularly beneficial in scenarios where multiple assessors are involved, as it promotes consistency and fairness in grading. The digital audit trail also allows for greater accountability, as all marking actions are recorded and can be reviewed if necessary.

The success of the RMA in catering to a wide range of rubric formats is largely attributed to its hybrid database storage approach. Although the application does not employ a traditional NoSQL database like MongoDB, it leverages the advanced features of the MariaDB database engine. This approach allows for the efficient storage and retrieval of unstructured data in JSON objects, providing the flexibility needed to handle diverse and complex rubric structures. This hybrid solution combines the reliability and robustness of a relational database with the flexibility of NoSQL databases, making it well-suited for the dynamic needs of educational assessments.

The core many-to-many relationship of the group module within the RMA is another significant feature, enabling its use beyond educational settings. This flexibility opens opportunities for the RMA to be adapted for various other applications, such as employee performance evaluations, peer reviews in professional settings, and other scenarios where structured assessment is required. The ability to manage complex relationships and hierarchies within the application ensures that it can scale and adapt to

different contexts, providing broad potential for future enhancements and integrations.

Stakeholder engagement and utilization of the RMA have been noteworthy. During the UAT phase, stakeholders, including administrators and assessors, provided valuable feedback that informed subsequent refinements to the application. Post-deployment, continued stakeholder involvement has been critical in assessing the RMA's impact and effectiveness. Regular training sessions and support resources have been made available to ensure that users can fully leverage the application's capabilities. Preliminary feedback indicates that stakeholders are actively utilizing the RMA and have integrated it into their routine assessment processes, contributing to a smoother and more efficient workflow.

While the RMA has proven effective in its current form, there are several avenues for future improvements. Enhancements could include the integration of advanced analytics to provide deeper insights into assessment data, the incorporation of artificial intelligence to assist in marking and feedback, and further customization options to cater to specific institutional needs. Additionally, expanding the application's compatibility with various educational technologies and platforms would further increase its utility and adoption.

In conclusion, the development and deployment of the Rubric Marking Application (RMA) represents a significant advancement in the field of educational technology. By addressing the challenges associated with traditional marking methods, the RMA offers a robust, transparent, and scalable solution for managing assessments. The successful implementation of hybrid database storage and the flexible group module design underpin the application's versatility and potential for broader applications. As educational institutions continue to embrace digital transformation, tools like the RMA will play a crucial role in enhancing the efficiency and fairness of the assessment process.

Acknowledgement

The authors would like to express their greatest appreciation to Politeknik Mukah (PMU) for the permission to carry out this development work. Finally, the authors would like to thank the anonymous reviewers for the feedback and Centre of Technology Development, Politeknik Mukah, for supporting this development project.

References

- Afrianto, I., Heryandi, A., & Finandhita, A. (2021). User acceptance test for digital signature application in academic domain to support the covid-19 work from home program. *IJISTECH*

- (*International Journal of Information System and Technology*), 5(3).
<https://doi.org/10.30645/ijistech.v5i3>
- Brookhart, S. M. (2013). *How to create and use rubrics for formative assessment and grading*. Ascd.
- Cattell, R. (2011). Scalable SQL and NoSQL data stores. *Acm Sigmod Record*, 39(4), 12-27.
- Chen, L., Wang, H., & Liu, S. (2021). Automated Grading System: Enhancing Efficiency and Accuracy in Educational Assessment. *International Journal of Artificial Intelligence in Education*, 31(2), 245-260.
- Cockburn, A., & Cockburn, L. (2008). *Writing effective use cases*. Pearson Education India.
- Elmasri, R. (2008). *Fundamentals of database systems*. Pearson Education India.
- Gandomi, A., & Haider, M. (2015). Beyond the hype: big data concepts, methods, and analytics. *International journal of information management*, 35(2), 137-144.
- García, E., Romero, C., & Ventura, S. (2019). Predicting Student Performance and Engagement in Online Learning Environments Using Unstructured Data. *Journal of Educational Data Mining*, 11(1), 1-20.
- Han, J., Haihong, E., Le, G., & Du, J. (2011, October). Survey on NoSQL database. In *2011 6th international conference on pervasive computing and applications* (pp. 363-366). IEEE.
- Kulak, D., & Guiney, E. (2012). *Use cases: requirements in context*. Addison-Wesley.
- Martillano, D. A., Chowdhury, A. F. D., Dellosa, J. C. M., Murcia, A. A., & Mangoma, R. J. P. (2018, November). Pindots: An assistive six-dot braille cell keying device on basic notation writing for visually impaired students with IoT technology. In *Proceedings of the 2018 2nd International Conference on Education and E-Learning* (pp. 41-47).
- Moniruzzaman, A. B. M., & Hossain, S. A. (2013). Nosql database: New era of databases for big data analytics-classification, characteristics and comparison. *arXiv preprint arXiv:1307.0191*.
- Panadero, E., & Jonsson, A. (2013). The use of scoring rubrics for formative assessment purposes revisited: A review. *Educational research review*, 9, 129-144.
- Reddy, Y. M., & Andrade, H. (2010). A review of rubric use in higher education. *Assessment & evaluation in higher education*, 35(4), 435-448.
- Sadalage, P. J., & Fowler, M. (2013). *NoSQL distilled: a brief guide to the emerging world of polyglot persistence*. Pearson Education.
- Sandin, E. V., Yassin, N. M., & Mohamad, R. (2016). Comparative Evaluation of Automated Unit Testing Tool for PHP. *International Journal of Software Engineering and Technology*, 3(2), 7-11.
- Santos, O. C., Boticario, J. G., & Barrera, C. (2015). Towards the validation of a data-driven learning architecture. *Journal of Universal Computer Science*, 21(7), 926-942.
- Sharma, V., Dutta, P., & Kaushal, R. (2015). Hybrid data management in educational databases using SQL and NoSQL. *Procedia Computer Science*, 70, 270-277.
- Smith, J., Brown, L., & Taylor, S. (2020). Integration of structured and unstructured data in an educational assessment platform. *International Journal of Educational Technology*, 25(4), 456-478.
- Smith, J., Johnson, A., & Williams, B. (2020). Enhancing Educational Assessment Through Integrated Structured and Unstructured Data. *Journal of Educational Technology & Society*, 23(3), 112-125.
- Sommerville, I. (2011). *Software engineering* (ed.). America: Pearson Education Inc.
- Stevens, D. D., & Levi, A. J. (2023). *Introduction to rubrics: An assessment tool to save grading time, convey effective feedback, and promote student learning*. Routledge.
- Stonebraker, M. (2010). SQL databases v. NoSQL databases. *Communications of the ACM*, 53(4), 10-11.
- Stonebraker, M., & Çetintemel, U. (2018). "One size fits all" an idea whose time has come and gone. In *Making databases work: the pragmatic wisdom of Michael Stonebraker* (pp. 441-462).
- Taipalus, T. (2023). Database management system performance comparisons: A systematic literature review. *Journal of Systems and Software*, 111872.