

Implementing Validated UI/UX Design into a Real System: An Agile Kanban-Based Development Evaluation in an Educational Information System

Indah Lestari^{1*}, M. Fakhri Salman², Anggy Trisnadoli³, Muhammad Mahrus Zain⁴

^{1,2,3,4} Information System Department, Faculty of Information Technology, Politeknik Caltex Riau, Pekanbaru, Riau, Indonesia

E-mail: ^{1*}indah@pcr.ac.id, ²fakhrisalman420@gmail.com, ³anggy@pcr.ac.id, ⁴mahrus@pcr.ac.id

(Received: 23 Jan 2026, revised: 27 Feb 2026, accepted: 3 Mar 2026)

Abstract

UI/UX research often ends at the prototype validation, while the implementation process remains limited in empirical studies. This study implements UI/UX designs validated in previous research into an information system and evaluates the role of the Agile Kanban approach in bridging the gap between design and real system implementation. The research method uses an implementation-based evaluation that includes system requirements alignment, Agile Kanban-based iterative development, and post-implementation evaluation. The results show that of the total 34 screens developed, 6 were implemented according to the initial design, 8 were adjusted during development, and 20 additional screens were developed to meet operational stakeholder needs. Development performance reached 135%, calculated as the ratio between completed and initially planned features. The development process was conducted through 4 Kanban cycles that allowed for controlled adjustments to business logic and feature additions. This study identified 5 main gaps in the implementation of the design in the real system. Post-implementation evaluation showed that all 79 functional requirements were successfully implemented with a total of 27 functional features, exceeding the initial plan of 20 features. The system was declared fully accepted through User Acceptance Testing and achieved a System Usability Scale score of 78.3 from 60 respondents, which is in the Acceptable category. These results indicate that Agile Kanban is effective in bridging the gap between UI/UX design validation and real system implementation, and is able to maintain usability quality in operational systems. This provides practical direction for organisations that seek to operationalise validated designs within dynamic development settings.

Keywords: UI/UX Implementation, Agile Kanban, Design-Implementation Gap, Usability Evaluation.

I. INTRODUCTION

User Interface and User Experience (UI/UX) design has a strategic factor in determining the success of modern information systems. UI/UX is not only a visual element aspect, but it also contributes to how users navigate the systems so they can understand system workflows and feel engaged. Some empirical studies have shown that good UI/UX design contributes significantly to improved usability, user satisfaction, and user engagement in digital systems, regardless of specific fields or case studies, whether in education, inventory, government, or business [1][2][3][4]. These findings confirm that UI/UX is a key factor directly influencing the acceptance and continued use of information systems.

As user expectations for intuitive and easy-to-use systems increase, research in the field of UI/UX is growing rapidly,

emphasising user-centred design [2], human-centred design [4], or design thinking [5] approaches, as well as identifying design factors that influence user experience. Literature reviews show that elements such as visual layout, information quality, navigation, high colour contrast, compatibility, responsiveness, and design consistency are important components in creating a positive user experience [4][6]. There are at least 43 factors that influence user experience [6]. In addition, UI/UX research also focuses heavily on analysing the relationship between interface quality and user satisfaction and user willingness to adopt technology through evaluative and quantitative approaches [7][8]. Good user interface design contributes significantly to user experience [3]. This shows that UI and UX are positively correlated with each other. Although providing a strong understanding of design aspects that influence user experience, most of these studies are still oriented towards evaluating existing



interfaces or systems. Most UI/UX research stops at the prototype stage, and the majority of design validation uses usability testing such as SUS, UEQ, and so on [1][6][7]. This validation is often considered sufficient to declare a design ready for implementation into software.

Although various UI/UX studies have demonstrated that designs validated through usability testing can improve interface quality and user experience, implementing these designs into real-world systems still presents challenges that have not been widely studied empirically. In software development practice, UI/UX designs validated at the prototype stage cannot always be directly implemented without significant changes or compromises. During the implementation phase, there are commonly challenges that frequently arise, including technical constraints, evolving requirements, operational and infrastructure issues, as well as development process dynamics that may affect the quality of the intended user experience.

Studies on large-scale system implementation evaluations indicate the existence of a design–reality gap between assumed design conditions and real-world system use, potentially influencing system success despite a well-developed initial design [9]. Furthermore, empirical findings on systems in real-world use also indicate that the quality of the user experience can be degraded by technical and contextual factors, such as system reliability, device compatibility, and accessibility limitations, which are not always identified at the prototype stage [10]. This situation indicates that UI/UX validation at the design stage is insufficient to guarantee the quality of the user experience in the final system. Therefore, research specifically evaluating the implementation of validated UI/UX designs in real systems, along with the associated challenges, is needed.

Therefore, research is needed to bridge the gap between UI/UX design validation and real-world system implementation. Furthermore, some studies have attempted to integrate User-Centred Design (UCD) into sequential development models like Waterfall, but found that the iterative nature of UCD is complex to accommodate within a rigid development flow, resulting in suboptimal design evaluation, implementation delays, and reduced user experience in the final system [11]. These findings point to the need for a flexible and adaptive development approach that helps align validated UI/UX designs with real-world system implementation, allowing both design and implementation to be refined together over the course of development.

On the other hand, studies on Agile transformation show that agile approaches, including Scrum and Kanban, emerged as responses to the shortcomings of traditional development methods, especially when dealing with uncertainty, frequent requirement changes, and fast-paced development environments. Kanban, in particular, emphasises workflow visualisation, flow management, and the flexibility of continuous priority adjustment, making it more adaptive to changes that arise during the implementation process [12]. Comparative studies of software development methodologies also confirm that Agile

approaches such as Scrum and Kanban have a much higher level of flexibility than sequential models, especially in the context of projects with dynamic requirements and involving continuous user feedback [13]. These findings indicate that an adaptive development approach is needed to bridge the gap between validated UI/UX designs and real-world system implementations, so that design and implementation iterations can run in harmony throughout the development cycle. Based on these considerations, this study adopts a Kanban-based Agile approach to implement UI/UX designs generated in previous studies into real-world information systems.

This research is a continuation of previous research that has produced a UI/UX design for an information system based on user-centred design and validated through usability testing at the prototype stage. Although the design has demonstrated good usability quality, previous research has not evaluated how the UI/UX design is implemented into a real system and how the development process affects the design realisation and the quality of the user experience in the final product. Therefore, this study aims to implement and evaluate the UI/UX design that has been produced in previous research [5] into the development of an information system using the Agile Kanban approach. Specifically, this study evaluates the role of Agile Kanban in supporting design-implementation iterations, managing feature priorities, and adapting to changing requirements during the development process, so that the quality of the designed user experience can be maintained in the implemented system. Based on the gap between UI/UX design validation at the prototype stage and the challenges of implementing it into a real system, and as a continuation of previous research that produced validated UI/UX designs, this study was formulated to answer the following research questions: (RQ1) How can UI/UX designs generated and validated in previous research be implemented into a real information system using the Agile Kanban approach? (RQ2) What is the role of Agile Kanban in supporting design-implementation iterations and managing feature priorities during the system development process that implements validated UI/UX designs? (RQ3) What are the levels of usability and system acceptance after implementing validated UI/UX designs in a real system?

The contributions of this study include (1) establishing empirical evidence regarding the implementation of validated UI/UX designs into a real system as a continuation of prototype-based research, (2) evaluating the effectiveness of the Agile Kanban approach in accommodating the iterative nature of UI/UX that is difficult to achieve in a sequential development model, and (3) presenting post-implementation evaluation results covering aspects of functionality, user acceptance, and system usability. Thus, this study extends the contribution of previous research from the prototype design and validation stage to the evaluation of implementation in operational systems.



II. RESEARCH METHODOLOGY

This section describes the research methodology, which is structured around an implementation-based evaluation approach that implements and evaluates the results of UI/UX designs validated in previous research [5] within a real

information system. The development process uses Agile Kanban because it supports iterative, adaptive development during implementation. The research stages are shown in Figure 1.

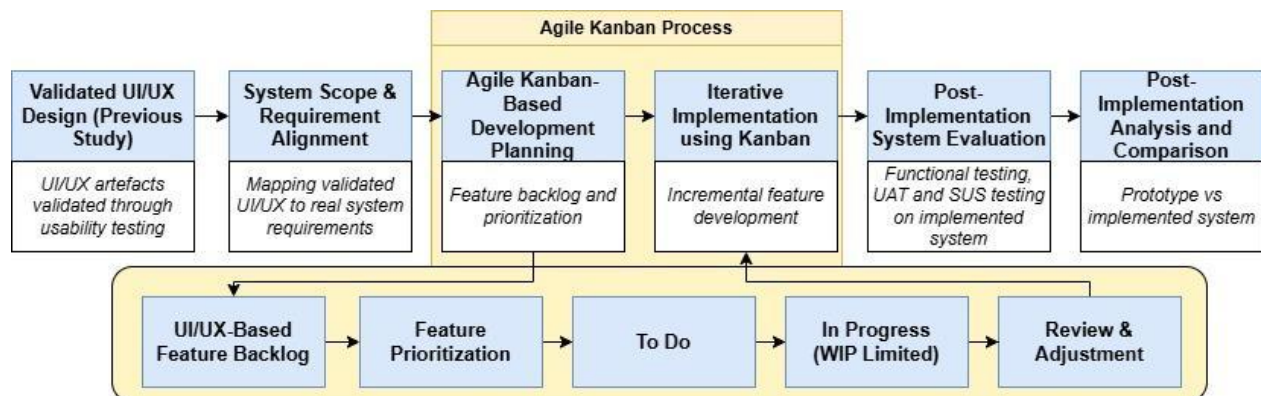


Figure 1. Research Methodology for Agile Kanban-Based Implementation

A. Validated UI/UX Design

This research does not design the entire display design from scratch, but instead emphasises the process of implementing existing UI/UX designs that have been validated for usability at the prototype stage [5]. Artefacts in the form of a Figma interface design and user interaction flow are input for this research.

B. System Scope & Requirement Alignment

The UI/UX design is not conceptually modified; it is aligned with the needs of the actual system implementation. At this stage, the design is mapped to the system's functional requirements. The technical and operational aspects of the system, as well as changes to business process flows based on updated stakeholder information, are considered. The outputs at this stage include as-is and to-be business process diagrams, use case diagrams, and ERDs.

C. Agile Kanban-Based Development Planning

Once system requirements are defined, the next stage is system development planning using the Agile Kanban approach, which supports flexible and adaptive progress, reassuring the audience of a responsive process. At this stage, all functional requirements derived from UI/UX design are organised into a UI/UX-based feature backlog. These features are then prioritised based on user value and implementation complexity. The development plan is not structured as a linear process, but rather as a list of features that can be implemented incrementally and adjusted if requirements change or new findings arise during the development process.

D. Iterative Implementation Using Kanban

The system implementation process is carried out iteratively using the Kanban mechanism. The workflow is visualised through a Kanban board consisting of To Do, In Progress, and Review and Adjustment stages. Each feature in

the In Progress stage is limited in number by applying a work-in-progress (WIP) limit to maintain workflow stability and prevent task accumulation. In the Review and Adjustment stage, implemented features are examined to confirm alignment between UI/UX design and system implementation. When gaps or required changes are identified, the features are returned to the backlog for further iteration, allowing design and implementation to progress in a coordinated and iterative manner.

E. Post-Implementation System Evaluation

Some evaluations are conducted to assess its quality in a real-world context following system implementation. These evaluations include functional testing to verify feature compliance, User Acceptance Testing (UAT) to examine user acceptance, and the System Usability Scale (SUS) to capture users' perceptions of usability. Combining these methods provides a comprehensive overview of system readiness, covering functionality, user acceptance, and usability aspects.

Functional testing covered 79 functional requirements across four user roles (General Users, Students, Teachers, and Admins). Each requirement was translated into structured test cases to verify input-output conformity, access control accuracy, and workflow consistency. Testing was conducted iteratively following each Kanban review cycle.

User Acceptance Testing (UAT) was conducted with the RTA secretary and administrative stakeholders. Acceptance criteria were based on scenario-based validation of operational workflows and feature conformity to institutional requirements.

SUS evaluation was conducted after the system passed UAT, with a representative method involving 60 respondents consisting of 35 students and 25 teachers. The standard 10-item SUS questionnaire with a 5-point Likert scale was used. The final SUS score was calculated using the standard

formula: (sum of adjusted item scores × 2.5), resulting in a score between 0 and 100.

F. Post-Implementation Analysis and Comparison

The analysis phase is conducted by reviewing the results of functional testing, user acceptance, and system usability after implementation. Post-implementation usability results are analysed and compared descriptively with the usability evaluation results from the previous UI/UX design phase to identify changes in the quality of the user experience. This analysis is used to evaluate the extent to which the validated UI/UX design implementation, supported by the Agile Kanban approach, is able to maintain design quality and meet user needs in the real system.

III. RESULTS AND DISCUSSION

This section presents the research findings and discussion based on the formulated research methodology, including the implementation of validated UI/UX design into a real-world information system, the role of the Agile Kanban approach during the development process, and post-implementation system evaluation. The discussion focuses on interpreting the research findings and their implications for bridging the gap between UI/UX design and real-world system implementation.

A. Implementation of Validated UI/UX Design Using Agile Kanban

The system developed is the Information System for the Rumah Tahfizh Akhwat Raudhatul Jannah (RTA) Women's Quranic Recitation Center. RTA, a women's Quranic education institution in Pekanbaru, previously operated manually, necessitating automation to manage the data of its student population of over 400.

This research addresses various technical and operational aspects, as detailed below:

1. Technical Aspects of Development
- a. Programming Languages: PHP and JavaScript.

b. Framework: Laravel to accelerate the development of application logic.

c. Database: MySQL for relational data management.

d. Platform: The system is built as a responsive website for access across multiple devices.

e. Management Tools: Using ClickUp to visualise Kanban boards and manage team workflows.
2. System Operations
- This system is designed to cover all RTA administrative operations by dividing access rights into four main actors:

a. General Users: Access institutional information and complete online registration.

b. Students: View tuition bills, upload proof of payment, monitor memorisation history, and register for exams.

c. Teachers: Input student memorisation deposit data directly into the system.

- d. Admins: Have full authority to manage registration data, confirm payments, manage donations, and schedule classes.

The system was implemented based on the UI/UX design from previous research, which consisted of 34 screens. The development results show that:

1. 6 screens were successfully implemented consistently according to the initial design, as shown in Figure 2.
2. 8 screens underwent minor adjustments during the development process at the request of stakeholders, as shown in Table 1 and Figure 3.
3. 20 additional screens were developed to accommodate 10 functional requirements that emerged during the system implementation phase, as shown in Table 2 and Figure 4. These were developed based on new operational requirements from stakeholders that were not yet included in the initial design. The developers maintained the consistency of typography and colour, matching the original design.

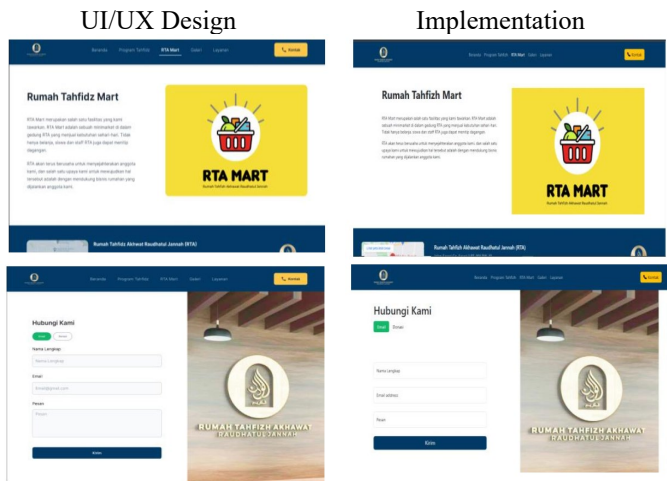


Figure 2. Some Views of Consistent Validated UI/UX Implementations

Table 1. List of Implementations with Minor Adjustments

Implementation	Change
Service Page (Login)	The "Student" or "Staff" login options in the initial design were changed to a unified login process for Students, Teachers, and Admins, and the RTA logo was added.
Student Summary Page (Dashboard)	Profile photos were removed (replaced with icons), Arabic welcome messages were added, class schedules and notifications were integrated directly into the dashboard, and exam registration buttons and password change features were added.
Student Payment Page	Billing notifications were moved to the dashboard; the "Upload" link was replaced with a "Pay" button.
Payment Receipt Upload Page	Student data is now automatically populated based on the logged-in



Implementation	Change
	account, so students only need to upload a photo of their proof of payment.
Student Exam Registration Page	Student personal data is automatically populated; students only need to add information about the number of juz' and tabi' categories.
Admin Dashboard Page	Metrics for the number of registrants, students, teachers, and suggestions were added, as well as annual payment charts and student data charts per class.
Admin Class Schedule Page	Admins were also added functionality to create new schedules and assign students to specific classes.
Admin Exam Registration Page	Data filtering (based on class/exam name) and exporting data to PDF and Excel formats were added.

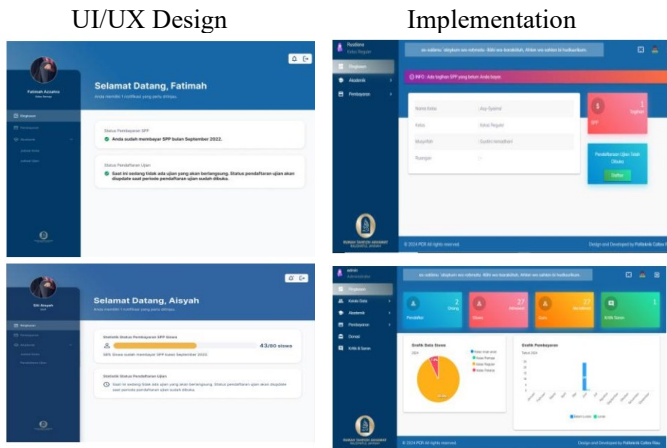


Figure 3. Some Validated UI/UX Implementation Views with Minor Changes

Requirements	Description
Donations	and date of the donation received
Criticism & Suggestions	Displays incoming messages from general users and features responding via email
Change Password	Security feature to change account password

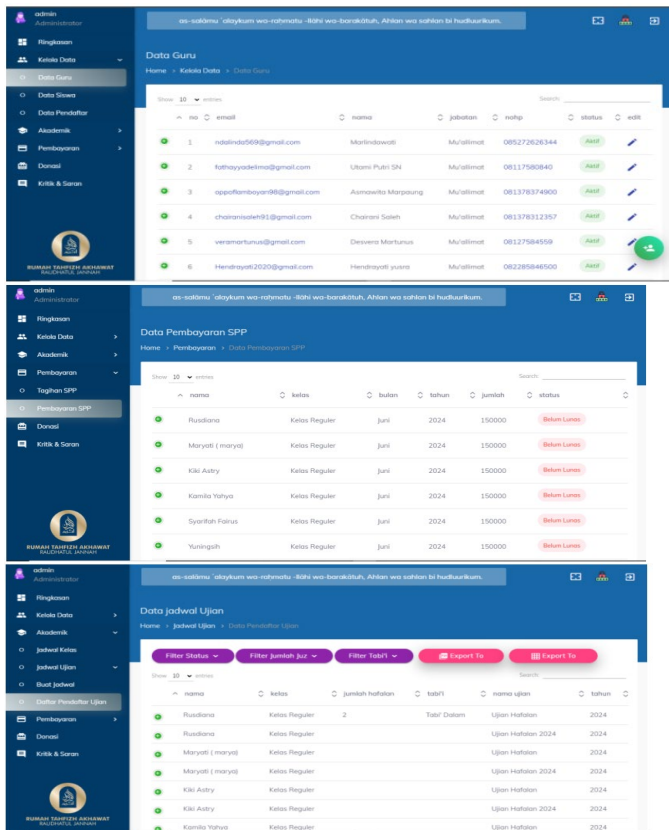


Figure 4. Some UI/UX Implementation Views of New Functional Requirements

Table 2. List of New Requirements Implementation	
Requirements	Description
Manage Teacher Data	Teacher registration, add, and edit data (email, position, phone number, etc.)
Manage Student Data	Active student registration, add, and edit data
Manage Registrants	Feature for admins to verify new registrants and change their status to "Active Student"
Class Details	Admin can add or remove students from a particular class.
Create Exam Schedules	The admin inputs the exam schedule which will later appear on the student dashboard
Admin Tuition Bills	Feature to add tuition bills in bulk based on class, month, and year
Manage Tuition Payments	The admin confirms and checks the student's proof of payment to change the status to "Paid"
Manage	Recording the name of the donor, amount,

During the implementation process using Agile Kanban, minor adjustments were made to align the system with real-world business processes better, while maintaining the aesthetics of the initial design. Feature expansion showed that the planned 20 features grew to 27 in the actual system. This indicates that there were no technical issues in implementing the validated UI/UX design. Challenges that arose were changes in business processes and new requirements that emerged during the development process. However, development using Agile Kanban, an iterative and adaptive development method, was able to accommodate these changes. Thus, UI/UX design implementation cannot be viewed as a linear process, but rather as an adaptive process that requires continuous adjustment. Therefore, the selection of a software development method significantly influences the success of the implementation. To clarify the relationship between interface artefacts and system functionality, Table 3 presents example mappings between screens, features, and functional requirements.

Table 3. Example Mapping of Screen, Feature, and Functional Requirement

Screen	Feature	Functional Requirement
Student Payment Page	Tuition Payment	FR12: Upload payment proof
Admin Dashboard Page	Monitoring	FR21: View Annual Payment Statistics
Donation Page	Donation	FR35: Manage Donation

B. The Role of Agile Kanban in Iterative Design and Implementation

In this study, Agile Kanban was used as the main driver to connect static UI/UX design with dynamic system implementation. Rather than serving solely as a task management tool, Kanban supported continuous iterations that allowed user needs to evolve throughout the development cycle. The following describes the role of Agile Kanban in supporting the design-implementation process:

1. Backlog Formation from UI/UX Design

The Backlog in this system did not emerge randomly, but was formed through two main pathways:

- Implementation of validated designs: All core features were derived from previous UI/UX designs using Design Thinking methods.
- Identification of Additional Requirements: New features (such as Manage Donations and Data Export) were developed after stakeholder meetings to address gaps between the initial design and actual operational needs.

2. Feature Prioritisation

The priority of the work was determined by analysing the correlation between User Value (Benefit) and Effort using the Eisenhower Matrix to divide features into four quadrants, as shown in Table 4. This determination was reached by the developers and stakeholders, resulting in the system implementation feature priorities shown in Figure 5.

Table 4. Feature Priority Based on User Value and Effort

Quadrant	Criteria	Example Features
Do Now	Important & Urgent	Home, Registration, Gallery, RTA Mart Submission
Do Next	Important, Not Urgent	Login, Teacher Data, Student Data, Memorisation
Do Later	Urgent But Less Important	Payment, Class Registration, Exam Registration
Do Last	Less Important and Less Urgent	Dashboard

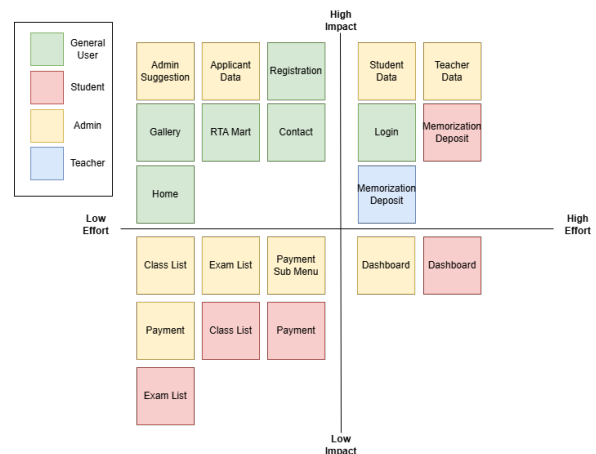


Figure 5. Feature Prioritisation

3. Kanban Workflow (*Visualising Work*)

To maintain transparency in the development process, the workflow is visualised using ClickUp, as shown in Table 5. Project management activities are supported through a Gantt chart visualisation in Figure 6, which uses colour indicators to represent feature priorities, ranging from urgent to low. The work visualisation is based on the Kanban columns, as shown in Figure 7.

Table 5. Workflow Configuration

Kanban Column	Function
To Do	Features selected based on priority
In Progress	Features under development (WIP-limited)
Review	Features requiring adjustment
Done	Features completed and validated

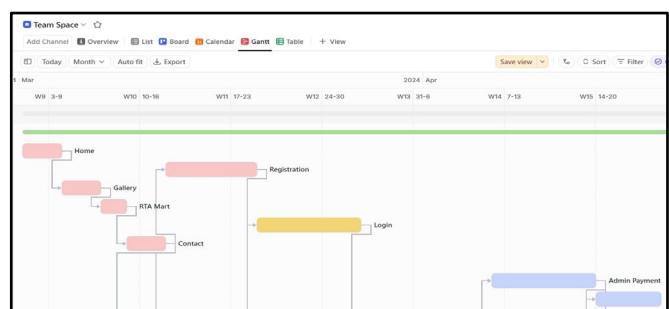


Figure 6. Ganttchart

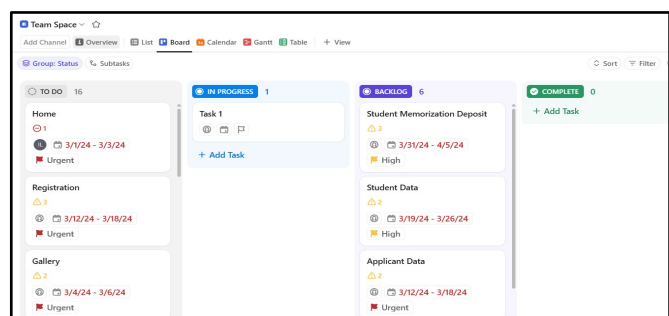
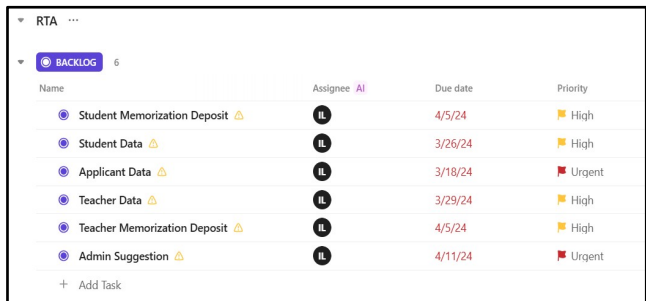


Figure 7. Visualize Work

4. The Strategic Role of WIP Limits and Reviews
- a. WIP (Work In Progress) Limit: Used to limit the number of ongoing tasks to maintain team focus and reduce wasted time. This allows developers to monitor due dates, as shown in Figure 8, closely. In the Phase 1 Meeting, the implementation of the work-in-progress limit proved effective, with the team successfully completing 7 features, exceeding the initial target of 5.



Name	Assignee	Due date	Priority
Student Memorization Deposit	IL	4/5/24	High
Student Data	IL	3/26/24	High
Applicant Data	IL	3/18/24	Urgent
Teacher Data	IL	3/29/24	High
Teacher Memorization Deposit	IL	4/5/24	High
Admin Suggestion	IL	4/11/24	Urgent

Figure 8. Limit Work in Progress

- b. Review & Adjustment: Meetings are held every 14 days with stakeholders to demonstrate progress. If a feature does not align with the business logic, it is moved back to the Backlog for adjustment, rather than being rushed to completion. The four Kanban review cycles (Table 6) demonstrate that continuous

- stakeholder feedback can be accommodated through iterative reviews without affecting system stability.
5. Impact on Flexibility and Stability The implementation of Kanban had a significant impact on system development performance, as shown in Table 7. Kanban implementation not only increased efficiency in completing core features but also enabled the faster and more efficient completion of additional work, including the following aspects:
- a. High Flexibility: Able to accommodate the addition of 20 features to 27 without excessively extending the project duration.
- b. Development Stability: Despite numerous changes, the system remained stable and was 100% validated through Black Box Testing at every stage.
- c. Time Efficiency: The project was completed in 53 days (8 days earlier than the initial plan of 61 days), improving performance by 135%. The development performance metric was calculated using the following Formula 1:

$$Performance\ (%) = \frac{\text{Total Completed Features}}{\text{Planned Features}} \times 100 \quad (1)$$

With 27 completed features compared to 20 initially planned features, the performance reached 135%. This metric reflects feature delivery effectiveness rather than time compression alone.

Table 6. Review and Iteration Process for System Development Using Kanban

Re view	Review Focus	Major Changes	Kanban Mechanisms Involved	Achievements
I	Initial implementation of core features and consistency with previous UI/UX designs	No significant logical changes; Implemented as designed	Focus on initial flow and completion of the prioritised backlog	Target of 5 features exceeded, resulting in 7 features
II	Validation of operational flows with stakeholders	Business logic changes: memorisation fees from per verse to per chapter (monthly); changes to the tuition management flow	Non-conforming features are returned to the backlog for revision	System adjustments to align with real-world needs
III	Evaluation of revised features and administrative requirements	Addition of new features: data export (PDF/Excel) and donation management; improved admin data filtering capabilities	Addition of new backlog and re-prioritisation of features	Administrative and reporting support expanded
IV	Final review and preparation for usability evaluation	Finalisation of the admin dashboard (student charts & payments); addition of a password change feature	Closing the iteration and stabilising the system	System completed with 27 functional features, meeting all stakeholder needs

Table 7. Evaluation of the Success of Implementation Using Kanban

Meeting	Feature Target	Feature Progress	Feature Completed	Performance	Target Duration	Completed
Meeting 1	4	7	7	35%	14 days	
Meeting 2	4	7	14	70%	14 days	45 days
Meeting 3	6	6	20	100%	17 days	
Meeting 4	6	7	27	135%	16 days	8 days
	20	27			61 days	53 days

C. Post-Implementation System Evaluation

A post-implementation system evaluation was conducted to assess the readiness of the Rumah Tahfizh Akhwat (RTA) Information System from three main aspects: system functionality, user acceptance, and usability. This evaluation was conducted after all features developed based on Agile Kanban had been implemented and the system was stable.

Functional testing demonstrated the system's readiness to support operations in terms of:

1. **Full Automation:** The system has proven ready to replace all RTA business processes previously performed manually.
2. **Workflow Stability:** Seven additional features submitted through the backlog have been successfully tested and show that the system architecture is stable enough to accommodate dynamic changes without disrupting existing functionality.
3. **Superior Performance:** Achieving performance improvements of up to 135% and completing the project 8 days ahead of schedule demonstrates that the system is not only technically functional but also efficient in its development process.

User acceptance testing, with results accepted without revision, demonstrated a strong correlation with the developer's implementation of the UI/UX design, including:

1. **Visual Consistency:** The developer successfully maintained high consistency with the original design [5], particularly on the public main page.
2. **Harmonisation of New Features:** Despite adding 20 screens beyond the initial design, the developer maintained consistent typography, colour, and design

principles, ensuring a uniform, professional user experience.

3. **Functional Adaptability:** High acceptance was also driven by the developer's willingness to revise eight main screens to align the system's logic with real-world business processes (such as changes to the memorisation deposit and tuition payment flows), making the system more intuitive for RTA staff.

Usability testing demonstrated the system's ease of use in the following aspects:

1. **Positive User Experience:** Scores demonstrated that the system was easy to learn and efficient for users in completing various routine tasks, such as tuition payments or memorising data.
2. **By applying the Agile Kanban approach,** this study successfully connected static design with real-world implementation. Although there are some gaps between static design and operational requirements, these were successfully bridged, resulting in an information system that was not only aesthetically pleasing in line with the initial UI/UX design but also highly functional for users.
3. **Successful Change Management:** Although the system expanded from an initial plan of 20 features to 27 functional features, this transition did not degrade the system's ease of use but instead strengthened its stability and utility for the organisation.

These results validated that the RTA information system was feasible and ready to replace the institution's management's manual business processes. A summary of the system evaluation results is shown in Table 8.

Table 8. Post-Implementation System Evaluation

Aspects	Methods	Key Indicators	Results	Implications
Functionality	Blackbox Testing	79 Functional Requirements, 4 Actors	100% successful	Operational Ready System
User Acceptance	UAT	Suitability to Stakeholder Needs	Accepted without revision	Eligible for Implementation
Usability	SUS	System Usability Score	- Students = 75.6 (Good) - Teachers = 81 (Excellent) - Average = 78.3 (Acceptable)	UX maintained post-implementation

D. Bridging the Gap Between UI/UX Validation and Real-System Implementation

Bridging the gap between UI/UX validation and real-system implementation is the core scientific achievement of this research. It demonstrates that even after a prototype has been validated in a controlled environment, the transition to a functional system requires a dynamic framework like Agile Kanban to address the "design-reality gap." Key findings indicate that validated designs do not necessarily translate into perfect real-world systems without adjustments. These

findings suggest that the most effective way to implement UI/UX is to treat validated designs as documents that Agile Kanban allows to evolve through real-world validation. A summary of the gaps is presented in Table 9. The five identified gap categories were derived from documented Kanban review cycles meeting minutes, backlog change logs in ClickUp, and iterative review records across four Kanban review cycles. Each gap was classified based on the type of change request, business logic adjustment, or additional requirement identified during implementation.

Table 9. Bridging the Gap Between UI/UX Validation and Real System Implementation

Gap Aspects	Research Findings	The Role of Agile Kanban	System Impact
Design and Real-World Needs	Initial design (20 features) does not address operational needs	Adaptive Backlog & Iteration	System was expanded to 27 features
UI Design and Business Logic	Business logic emerges during implementation	Periodic Backlog Review & Revision	System flow is more intuitive
Prototype vs. Real-World System	System requires additional administrative features	Incremental Prioritization	System is operationally ready
Aesthetics vs. Stability	Risk of design inconsistency	Iteration & WIP Control	UI/UX consistency maintained
Usability Design vs. Implementation	Risk of reduced usability	Controlled Iteration	SUS score remains high

IV. CONCLUSION

This study shows that a UI/UX design validated at the prototype stage can be effectively implemented in a real-world information system using Agile Kanban. However, the findings indicate that design validation alone is insufficient to ensure system success, as the implementation process must respond to real-world requirements, business logic, and organisational dynamics.

The use of Agile Kanban effectively supported an iterative and adaptive implementation process by enabling systematic backlog management, clear feature prioritisation, and ongoing stakeholder review. As a result, the system was able to evolve considerably without compromising the design's consistency or the usability quality of the previously validated system.

Post-implementation evaluation results based on 3 aspects, specifically functionality, user acceptance and usability, show that the system meets functional requirements, achieves user acceptance without revisions, and demonstrates good usability. These results indicate that adaptive development can successfully bridge the gap between UI/UX validation and real-world system implementation. Thus, this study extends the contribution of UI/UX research from the prototype design and validation stages to implementation evaluation in real-world operational systems.

Despite these findings, this study has several limitations. First, it was conducted as a single case study within one educational institution, which may limit generalizability to other domains. Second, stakeholder involvement in UAT may introduce acceptance bias. Third, the development team's familiarity with Agile Kanban may influence performance outcomes. To mitigate these risks, structured documentation, iterative validation, and standardised usability instruments were applied throughout the development process.

REFERENCES

[1] T. I. Hasan, C. I. Silalahi, R. Y. Rumagit, and G. D. Pratama, "UI/UX Design Impact on E-Commerce Attracting Users," in *Procedia Computer Science*, Elsevier B.V., 2024. doi: <https://doi.org/10.1016/j.procs.2024.10.336>.

[2] P. M. Effendi and Endra Rahmawati, "Peningkatan Terhadap Pengalaman Pengguna: Implementasi Pendekatan User-Centered Design dan Evaluasi Usability pada Prototipe Sistem Inventori," *INFOTECH: Jurnal Informatika & Teknologi*, vol. 6, no. 2, 2025, doi: <https://doi.org/10.37373/infotech.v6i2.1994>.

[3] K. Abertho, E. Dwinata, A. Gustho, R. A. Prasetyo, and W. E. Jayanti, "Analisis Pengaruh User Interface Terhadap Penggunaan Aplikasi Pembelajaran," *Jurnal Teknologi Sistem Informasi*, vol. 6, no. 1, 2025, doi: <https://doi.org/10.35957/jtsi.v6i1.10542>.

[4] A. Fritz Wijaya, T. Matius Surya Mulyana, and G. Janico Liunard, "Desain User Experience (UX) Pada Website Smart City untuk Meningkatkan Aksesibilitas Layanan Publik," *Jurnal Teknologi dan Manajemen Industri Terapan (JTMIT)*, vol. 4, no. 2, 2025, doi: <https://doi.org/10.55826/jtmit.v4i2.582>.

[5] A. A.-Z. Ibrahim and I. Lestari, "Perancangan UI/UX Pada Website Rumah Tahfidz Akhwat Menggunakan Metode Design Thinking," *Teknika*, vol. 12, no. 2, 2023, doi: <https://doi.org/10.34148/teknika.v12i2.599>.

[6] Sudiana, Y. U. Chandra, and L. Angela, "Key success factors for a better user experience in e-commerce website," in *Proceedings of 2021 International Conference on Information Management and Technology, ICIMTech 2021*, Institute of Electrical and Electronics Engineers Inc., 2021. doi: <https://doi.org/10.1109/ICIMTech53080.2021.9535076>.

[7] A. Jamal and S. Hashmat, "Innovations in UI/UX Design Of Mobile Applications: Trends, Practices And Challenges," *Spectrum of Engineering Sciences*, vol. 3, no. 5, 2025, doi: <https://doi.org/10.5281/zenodo.15478961>.

[8] L. Judijanto and D. Wardhani, "The Impact of Mobile Payment Systems and User Interface Design on Customer Adoption and Transaction Volume in E-commerce Companies in Jakarta Article Info," *The Eastasouth Journal of Information System and Computer Science*, vol. 2, no. 01, 2024, doi: <https://doi.org/10.58812/esiscs.v2i01>.

[9] F. G. Gultom and T. M. Y. Khudri, "Evaluasi Implementasi Sistem Aplikasi Keuangan Tingkat



- Instansi (SAKTI) Dengan Metode Design-Reality Gap,” *Owner: Riset & Jurnal Akuntansi*, vol. 8, no. 3, 2024, doi: <https://doi.org/10.33395/owner.v8i3.2127>.
- [10] M. Mujiono, T. O. Wibowo, and S. Alvin, “An Exploration of University Students’ Experience Using UI-UX Based E-learning Platforms: Challenges and Recommendations,” *International Journal of Social Learning (IJSL)*, vol. 5, no. 2, 2025, doi: <https://doi.org/10.47134/ijsl.v5i2.360>.
- [11] S. R. Wulan, Z. P. Hamida, A. Kurniaty, and M. R. Alifi, “Implementation of Waterfall Model and User-Centered Design Integration,” *INTEGER: Journal of Information Technology*, vol. 9, no. 2, 2024.
- [12] O. Mojabi, “Harmonizing Agile Transformation: Navigating Software Startup Success Through A Hybrid Journey of Scrum and Kanban Methods,” Blekinge Institute of Technology, 2024.
- [13] R. E. Hidayah and R. Agus, “Analisis Kebutuhan Kerangka Kerja Adaptif Berdasarkan Studi Komparatif Metodologi Pengembangan Perangkat Lunak,” *Technologia : Jurnal Ilmiah*, vol. 17, no. 1, 2026, doi: <https://doi.org/10.31602/tji.v17i1.20612>.