Development of an Android-Based Financial Management Information System for Boarding Houses Using the Rapid Application Development (RAD) Method

Feliks Prasepta Sejahtera Surbakti [1], Cynthia Chabella [1], Henry Edison [2]

[1] *Industrial Engineering Department, Atma Jaya Catholic University of Indonesia, Jakarta, Indonesia*
[2] *Software Engineering Department, Blekinge Institute of Technology, Karlskrona, Sweden*

Abstract **– The rapid advancement of technology has significantly transformed daily human life, particularly through the widespread use of smartphones to simplify various tasks. In response to this technological shift, Kost XYZ, a women's boarding house located in Central Jakarta, aims to enhance its operational efficiency. The boarding house faces challenges such as varying room facilities, inconsistent rental rates, non-permanent tenants, and manual record-keeping, all of which frequently result in delayed payments. To address these issues, this study proposes the development of an Android-based financial management information system using the Rapid Application Development (RAD) methodology. The system was built using Kotlin programming language via Android Studio and integrated with Firebase for data storage. Its design provides separate interfaces for owners and tenants: the owner system focuses on boarding house operations, while the tenant system facilitates payment management. Usability testing was conducted using five metrics-learnability, efficiency, memorability, error rate, and satisfaction with direct user involvement.**

The results indicate that the system is feasible and performs well across all usability aspects. A satisfaction score of 73.5 places the system in the "good" category, demonstrating user acceptance and the system's effectiveness in supporting real-time operational and financial monitoring.

Keywords **– Boarding house, information system, rapid application development, Kotlin.**

DOI: 10.18421/TEM144-36
https://doi.org/10.18421/TEM144-36

Corresponding author: Feliks Prasepta Sejahtera Surbakti, *Industrial Engineering Department, Atma Jaya Catholic University of Indonesia, Jakarta, Indonesia*
Email: feliks.prasepta@atmajaya.ac.id

Received: 05 May 2025.
Revised: 25 September 2025.
Accepted: 04 October 2025.
Published: 27 November 2025.

1. Introduction

Over time, human life has undergone significant transformations, driven by advances in technology and science. One prominent example is the increasing adoption of smartphones, which provides convenience through their portability and practical functionality [1]. Kost XYZ, a women's boarding house located in Central Jakarta, is also adapting to these modern developments. The rooms at Kost XYZ vary in size and amenities, leading to differences in rental fees. These variations often result in delayed payments, negatively impacting the boarding house's operations. This issue stems from the owner's difficulty in monitoring payments accurately, particularly regarding rental prices and payment deadlines, as well as the continued use of manual payment records. To address these challenges, an information system can be implemented as a solution.

An information system can provide accurate data to streamline operations, enhance efficiency, improve quality while minimizing costs, support decision-making processes, aid in planning and control, as well as facilitating internal control mechanisms [2]. A mobile-based system is preferred to increase accessibility through smartphones without requiring a PC [3]. Furthermore, mobile information systems offer advantages such as easy accessibility, portability, offline functionality, real-time notifications and interactions, in addition to access to device features [4].

Given these benefits, designing an information system serves as a viable solution for addressing the financial management issues at Kost XYZ.

The system will be developed for the Android platform. It is expected that this system will enhance the financial management of Kost XYZ by enabling real-time and accurate operational record-keeping. The Android-based system, which leverages the portability of smartphones, is anticipated to be user-friendly and practical. Moreover, it is expected to simplify payment monitoring, encouraging tenants to pay rent on time.

2. Methodology

The research methodology serves as a conceptual framework that adheres to established rules, activities, and procedures to facilitate the execution of the research process. This research employs the Rapid Application Development (RAD) methodology, as illustrated in Figure 1. RAD is a system development model characterized by short development cycles and incremental delivery [5]. One of the main advantages of RAD is its efficiency in terms of time, resources, and cost in the development of information systems [6]. The RAD model consists of three main phases: requirements planning, RAD design workshop, and implementation.

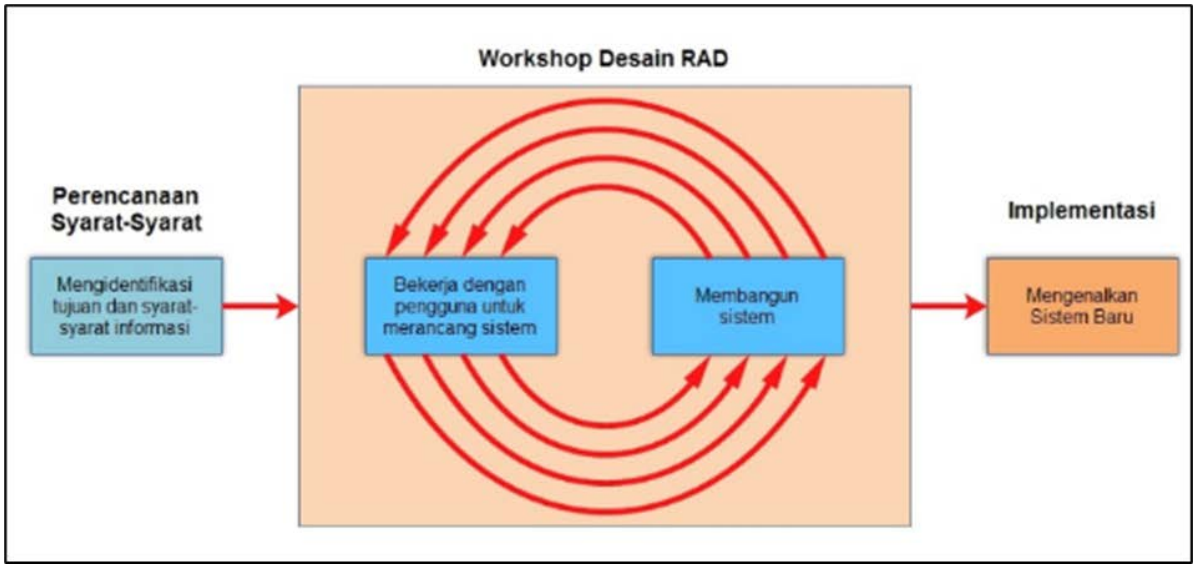

Figure 1. Illustration of the RAD model phases

2.1. Requirement Planning

Requirements planning involves data collection activities. The data collection process consists of three main steps: identifying the current system, identifying information needs, and identifying users. The identification of the current system is carried out through interviews with the boarding house owner to understand the existing payment process. The identification of information needed includes analyzing the required information, the parties who need and provide it, the timing of the events, the current system conditions, and the consequences arising from the current system [7]. Meanwhile, the user identification stage outlines the system's users in order to determine their roles and authorities within the system.

2.2. RAD Design Workshop

Unified Modeling Language (UML) is an object-oriented modeling technique used in software development. It is a standardized language designed to specify, visualize, construct, and document the artifacts of a software system [8]. In the RAD design workshop phase, UML is utilized through several diagrams, including class diagrams, use case diagrams, activity diagrams, and sequence diagrams [9].

This phase also includes the design and construction of the system. The system is developed using the Kotlin programming language within the Android Studio environment and is integrated with Firebase as the data storage platform. The system is designed based on user requirements to ensure alignment with their needs. After the coding process using Kotlin, the next step is to run the system to verify its functionality [10]. This is done using the Android Virtual Device (AVD) emulator available in Android Studio to simulate and test the system's performance.

2.3. Implementation

Implementation is the process of introducing the system to its users. Implementation is carried out through testing with end users. The testing phase utilizes acceptance testing in the form of usability testing. Usability testing aims to assess whether the developed system can be easily used, learned, and remembered by users. The testing is conducted with actual users, specifically the boarding house owner and the tenants. According to [11], usability testing it can be performed with a sample of 3–5 respondents; in this study, the system was tested with four respondents. The usability testing focused on five key aspects: learnability, efficiency, memorability, error rate, and user satisfaction [12].

2.4. Learnability

Learnability refers to the ease with which users can use the system. To assess this, users are given specific tasks to measure how easily it is possible to complete them. The evaluation of learnability is based on the success of respondents in carrying out these tasks. There are three success categories: success (S), partial success (P), and failure (F) [13].

The next step involves calculating the success rate. According to [14], the average task completion rate for a system is 78%. Therefore, a system is considered to have good learnability if it achieves a success rate above this average. The formula for calculating the success rate is as follows [15]:

$$success\ rate = \frac{success\ rate + (partial\ success\ task\ \times 0.50)}{total\ task} \quad (1)$$

2.5. Efficiency

Efficiency is used to measure the speed at which a user performs a task. To assess this, users are given specific tasks, and the time taken to complete each task is recorded to evaluate how quickly or slowly users can complete a given process. The efficiency test is conducted using a stopwatch to measure the task completion time. The recorded times are then used to calculate time-based efficiency using the following formula [16]:

$$Time\ Base\ Efficiency = \frac{\sum_{j=1}^{R}\sum_{i=1}^{N}\frac{n_{ij}}{t_{ij}}}{NR} \quad (2)$$

2.6. Memorability

Memorability measures the system's ability to be easily remembered by users, both by first-time users and those who have not used the information system for an extended period.

This evaluation is conducted by comparing the time required to complete a task on the first day with the time taken after a certain interval.

A system with high memorability allows users to re-familiarize themselves quickly without needing to relearn the interface. This is particularly important for systems that are used intermittently, as it indicates how intuitive and consistent the user interface design is over time.

2.7. Error

Error refers to the mistakes made by users while performing assigned tasks. The evaluation involves counting the number of errors committed by respondents during the task execution. Subsequently, the error rate is calculated to assess the system's accuracy. According to [17], the average number of errors per task is 0.7. A lower error rate indicates a more reliable and user-friendly system. High error rates may suggest issues with the system's interface design, task complexity, or lack of user guidance, and may necessitate further refinement to enhance usability.

2.8. Satisfaction

Satisfaction measures user satisfaction with the developed system through the use of questionnaires. This evaluation employs the System Usability Scale (SUS), which utilizes a Likert scale ranging from 1 (strongly disagree) to 5 (strongly agree). The resulting SUS score is then multiplied by 2.5, followed by the calculation of the average score. The average score is subsequently categorized according to the Sauro-Lewis Curved Grading Scale (CGS), as presented in Table 1. A higher average score indicates a greater level of user satisfaction and reflects positively on the system's usability and acceptance. The SUS method is widely recognized for its reliability in evaluating subjective user perceptions across different types of systems [18].

Table 1. Sauro Lewis CGS

SUS	*Grade*	*Percentile Range*	*Adjective*	*Acceptable*	*NPS*
84.1 - 100	A+	96-100	*Best Imaginable*	*Acceptable*	*Promoter*
80.8 - 84.0	A	90-95	*Excellent*	*Acceptable*	*Promoter*
78.9 - 80.7	A-	85-89	*Good*	*Acceptable*	*Promoter*
77.2 - 78.8	B+	80-84	*Good*	*Acceptable*	*Passive*
74.1 - 77.1	B	70-79	*Good*	*Acceptable*	*Passive*
72.6 - 74.0	B-	65-69	*Good*	*Acceptable*	*Passive*
71.1 - 72.5	C+	60-64	*Good*	*Acceptable*	*Passive*
65.0 - 71.0	C	41-59	*Ok*	*Acceptable*	*Passive*
62.7 - 64.9	C-	35-40	*Ok*	*Marginal*	*Passive*
<51.7	D	15-34	*Ok*	*Marginal*	*Detactor*

3. Results and Discussion

The following describes the stages of the current payment process implemented at Kost XYZ. The flowchart illustrates two main entities involved in the rental payment process: the boarding house owner and the tenant. There are two types of payment processes: one that involves billing on or after the due date, and one without billing, which typically occurs before the due date. The payment process involving billing proceeds as follows:

1. The boarding house owner checks the payment due date, which is based on the tenant's initial check-in date.
2. If the current date has reached the due date, the owner refers to the amount recorded in the manual logbook and issues a payment request to the tenant.
3. The tenant proceeds with the payment.
4. The boarding house owner receives and confirms the payment.
5. The boarding house owner records the payment date manually in the logbook.

The information system for Kost XYZ was developed using the Kotlin programming language, with the assistance of Android Studio as the development environment. The system design and construction were based on the identified needs of the tenants and the user interface designs created using Figma. Data storage for the Kost XYZ information system utilizes Firebase, which serves as a real-time database and also functions as the user authentication system. The following presents the Kost XYZ information system.

3.1. Tenant Information System

The following is the design of the tenant information system, which consists of the login and registration page, home page, profile page, transaction status page, and payment upload page.

The login page is the initial interface displayed upon launching the application. On this page, tenants are required to enter their email and password to gain access to the system. The login feature is restricted to tenants who have previously registered an account, as depicted in Figure 2. Therefore, tenants who do not yet have an account may select the "register now" option, which will redirect them to the registration page.

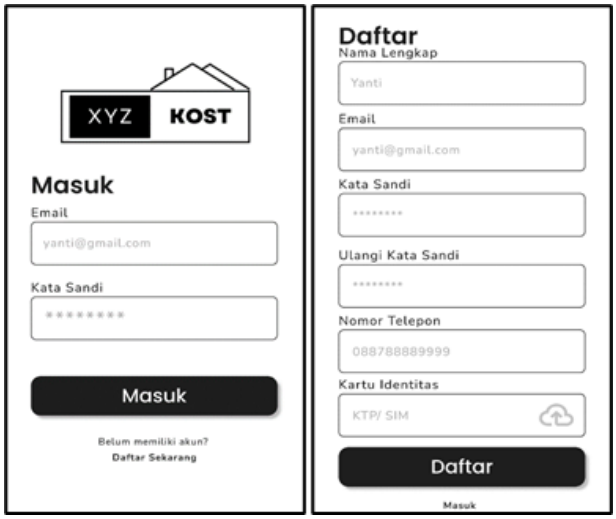

Figure 2. Tenant login page

The home page is designed as the central hub of the tenant information system for Kost XYZ. Tenant login page can be seen in Figure 3. This page presents key information such as “My Bills”, which displays the tenant’s current payment obligations, and “Transaction Status”, which provides details regarding the tenant’s past and ongoing transactions. In addition, tenants can access their profile from this page to view detailed account information.

Figure 3. Tenant home page

The profile page is presented when the tenant selects the profile option from the home page, as shown in Figure 4. This page is designed to provide tenants with access to their account information as submitted during registration, including email address, phone number, date of joining, and identification card details. Additionally, this page allows tenants to log out of the system.

Figure 4. Profile page

The transaction status page, as illustrated in Figure 5, is designed to display information regarding monthly payment status. Status indicators are differentiated by color codes: red indicates unpaid, green indicates paid, blue indicates under verification, yellow indicates invalid payment, and grey indicates that the due date has not yet been reached. The use of distinct colors is intended to help tenants easily interpret their transaction status while using the system.

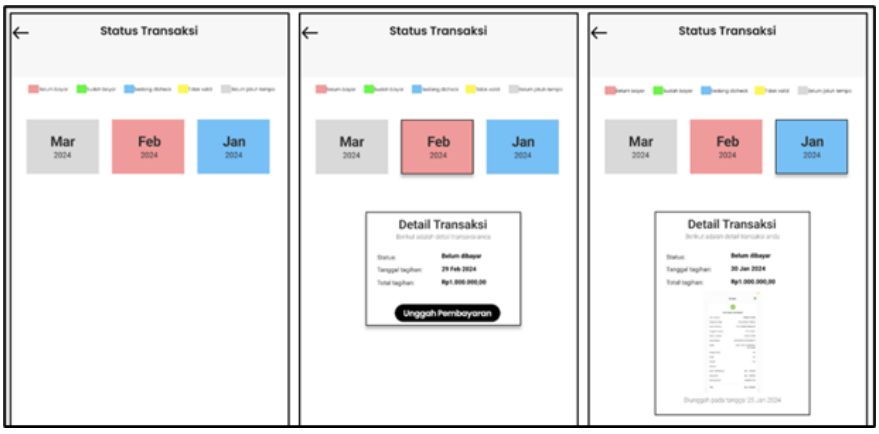

Figure 5. Transaction status page

The payment upload page is designed to assist tenants in submitting proof of payment. This page is accessed when the tenant selects the "Upload Payment" option on the Transaction Status Page. It provides a summary of essential payment information, including the payment due date and the total amount due, to facilitate accurate and timely payment submissions. Additionally, the page incorporates user-friendly instructions to minimize errors during the upload process. A confirmation message is displayed upon successful upload to ensure that tenants are informed of the completion status. The payment upload page is illustrated in Figure 6.

Figure 6. Payment upload page

3.2. Owner Information System

The following is the design of the owner information system, which consists of the Login and Registration Page, Home Page, Payment Summary Page, Manage Boarding House Page, Room and Tenant Details Page, and Edit Room Page.

The login page serves as the initial interface upon launching the application. On this page, the owner is required to enter a username and password to gain access to the system. In terms of interface design, the owner's login page shares a similar layout with that of the tenant. However, the registration page for owners differs in that it does not require identity card information during the registration process.

The owner's home page is displayed once the owner successfully logs into the system. On this page, the owner can perform payment summaries and manage the boarding house. The owner home page is illustrated in Figure 7.

Figure 7. Owner home page

The payment summary page is designed to provide an overview of boarding house payment transactions. On this page, the owner can view the payment status, total amount, due date, and necessary actions for each tenant. The available actions include verification, billing, warning, and reminding. The Payment Summary Page is presented in Figure 8.

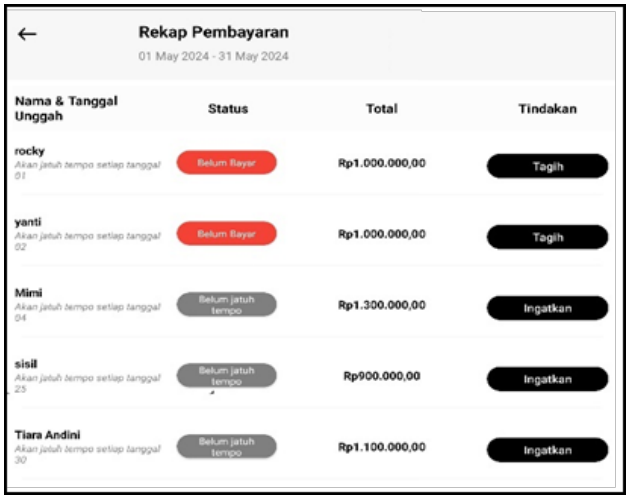

Figure 8. Payment summary page

The Manage Boarding House page functions to display the list of rooms available in the boarding house. On this page, the owner can add, modify, and view rooms as well as view tenant information. In the lower left corner, there is an add (+) button, which serves as the feature for adding new rooms. The add room function allows the owner to create and register new rooms according to the number of available units. The boarding house management page is illustrated in Figure 9.

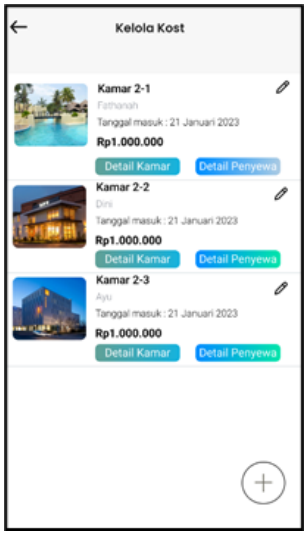

Figure 9. Boarding house page management

The room details page, as illustrated in Figure 10, is designed to display information related to the room, such as the room name, type, and rental price. Meanwhile, the tenant details page enables the owner to view tenant identity information, including the tenant's full name, email address, phone number, and identification card details.

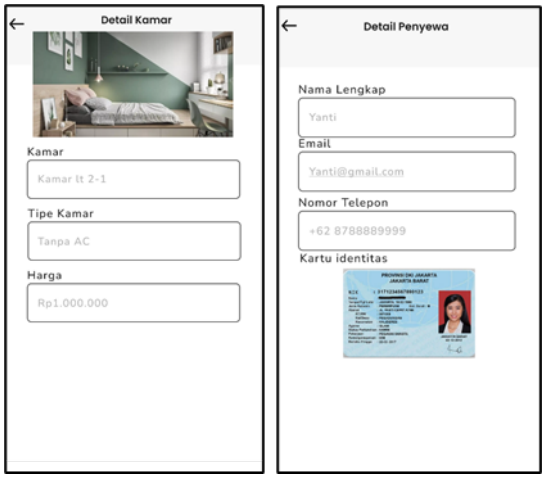

Figure 10. Room and tenant details page

The Edit Room page (Figure 11) allows users to modify, update, or delete room-related data. Examples of editable information include the room name, room type, rental price, and tenant assignment.

Figure 11. Edit room page

3.3. Testing

After the development of all information system pages were completed and confirmed to function according to the design, the next step was to conduct system testing. The information system was evaluated using usability testing. The usability testing focused on several key aspects: learnability, efficiency, memorability, error, and satisfaction. The usability testing for the tenant information system involved four respondents, consisting of tenants from Kost XYZ and the boarding house owner (Table 2 and Table 3).

Table 2. Task codes for tenant information system testing

Task Code	Task
T1	Register as a system user
T2	Log in to the system
T3	Make a payment
T4	View payment history
T5	View profile (identity card)
T6	Activate payment notifications

Table 3. Task codes for owner information system testing

Task Code	Task
T1	Register as a system user
T2	Log in to the system
T3	Add room data
T4	Assign tenant name to a room
T5	Issue a boarding fee invoice
T6	Activate payment notifications

Learnability refers to the respondents' success in completing specific tasks. The learnability evaluation is categorized into three levels of success: success (S), when the task is completed without any errors; partial success (P), when the number of errors is less than or equal to the total number of steps; and failure (F), when the number of errors exceeds the total number of steps.

The following Table 4 and Table 5 present the success rates for the tenant and owner information systems.

Table 4. Success rate of the tenant information system

Respon-dent	Task					
	T1	T2	T3	T4	T5	T6
1	S	S	S	S	S	S
2	S	S	S	S	S	S
3	S	S	S	S	S	S
4	S	S	S	S	S	S

Table 5. Success rate of the owner information system

Trial	Task					
	T1	T2	T3	T4	T5	T6
1	S	S	S	S	S	S
2	S	S	S	S	S	S
3	S	S	S	S	S	S
4	S	S	S	S	S	S

Based on the success rate calculations, the learnability evaluation results showed a score of 100% for the owner information system and 98% for the tenant information system. These results indicate that the developed information systems perform well, as exceeding the average task completion benchmark of 78% [19].

Efficiency aims to measure the speed at which users complete tasks by evaluating the time taken to finish a given process. Efficiency testing is conducted using a stopwatch to record the task completion time. Tasks completed without errors are assigned a score of 1, whereas tasks with errors are assigned a score of 0. The results of the tenant information system efficiency in Table 6 show that all tasks were completed without errors. The task completion time data for the tenant information system can be seen in Table 7. Meanwhile, for the owner information system efficiency results, there was one task performed with an error, as shown in Table 8. The task completion time data for the owner information system can be seen in Table 9. Prior to the calculations, both tenants and owners assumed that completing a single task would require approximately one minute, equivalent to 0.0167 goals per second.

Table 6. Efficiency results of the tenant information system

Respon-dent	Task					
	T1	T2	T3	T4	T5	T6
1	1	1	1	1	1	1
2	1	1	1	1	1	1
3	1	1	1	1	1	1
4	1	1	1	1	1	1

Table 7. Task completion time data for tenant information system

Respon-dent	Task (in seconds)					
	T1	T2	T3	T4	T5	T6
1	58.86	12.53	26.29	7.92	4.06	6.64
2	82.99	17.02	32.25	9.45	4.93	5.35
3	60.23	13.46	29.67	8.34	4.78	5.81
4	65.94	12.87	25.96	8.83	4.47	6.06

Table 8. Efficiency results of owner information system

Trial	Task					
	T1	T2	T3	T4	T5	T6
1	1	1	1	0	1	1
2	1	1	1	1	1	1
3	1	1	1	1	1	1
4	1	1	1	1	1	1

Table 9. Task completion time data for the owner information system

Respon-dent	Task (in seconds)					
	T1	T2	T3	T4	T5	T6
1	73.53	19.39	33.28	25.12	14.84	23.67
2	69.78	20.77	35.9	23.89	13.54	22.10
3	70.07	18.99	31.58	23.53	15.35	23.41
4	73.75	19.2	30.23	24.11	15.03	21.62

The time-based efficiency result for the tenant was 0.10 goals/sec. This result indicates a task completion speed of 0.10 goals/sec. The achieved result of 0.10 goals/sec exceeds the tenant's expectations. Similarly, for the owner information system, the result obtained was 0.04, which also surpasses the expected value. Therefore, both the tenant and owner information systems meet the efficiency criteria based on the testing conducted. The results pertaining to task completion time and efficiency for the owner and tenant information system are presented in Table 10 and Table 11.

Table 10. Comparison of task completion time for the tenant information System

Respondent	**T1**		**T2**		**T3**		**T4**		**T5**		**T6**	
	The nth day		The nth day		The nth day		The nth day		The nth day		The nth day	
	1	7	1	7	1	7	1	7	1	7	1	7
1	58.86	55.59	12.53	10.35	26.29	21.50	7.92	7.89	4.06	3.95	6.64	5.95
2	82.99	80.57	17.02	12.42	32.25	27.59	9.45	9.26	4.93	4.14	5.35	4.65
3	60.23	59.75	13.46	8.58	29.67	28.08	8.34	7.13	4.78	4.23	5.81	4.14
4	65.94	62.79	12.87	9.07	25.96	25.20	8.83	7.26	4.47	4.15	6.06	5.92

Table 11. Comparison of task completion time for the owner information system

Trial	**T1**		**T2**		**T3**		**T4**		**T5**		**T6**	
	The nth day		The nth day		The nth day		The nth day		The nth day		The nth day	
	1	7	1	7	1	7	1	7	1	7	1	7
1	73.53	70.53	19.39	17.56	33.28	31.65	25.12	23.43	14.84	13.21	23.67	22.41
2	69.78	68.87	20.77	16.97	35.9	30.78	23.89	22.90	13.54	12.81	22.10	21.73
3	70.07	66.82	18.99	17.02	31.58	30.19	23.53	22.11	15.35	12.76	23.41	21.24
4	73.75	68.62	19.2	17.60	30.23	31.40	24.11	21.87	15.03	12.09	21.62	21.18

Memorability measures the system's ability to be easily remembered by both new users and those who have not interacted with the information system for a certain period. The memorability testing was conducted on the seventh day after the initial testing. Based on Table 10 and Table 11, a comparison between the task completion times on the first and seventh days can be observed. The tables show that all tasks on the seventh day were completed more quickly than on the first day. These results indicate that the developed information system is memorable for both owners and tenants, even after several days of non-use. Furthermore, the results demonstrate relatively faster completion times compared to the first day, suggesting that users had become more familiar with performing tasks within the system.

Errors refer to mistakes made by users while performing tasks according to the predefined scenarios. This test measures the number of errors made by the respondents. In the error testing, the number of errors (F) is recorded for each step (K) within the task process.

Based on the calculation results, the error rate value obtained is lower than the average acceptable error threshold per task, which is 0.7 [20]. A lower error rate indicates a better-performing system. Based on the error rate results, the XYZ boarding house information system meets the criteria, as it falls below the 0.7 threshold.

Therefore, both the tenant and owner information systems of XYZ boarding house pass the error testing evaluation. The error results for the Tenant and Owner Information System are presented in Table 12 and Table 13.

Table 12. Tenant information system error

Respon-dent	T1		T2		T3		T4		T5		T6	
	F	K	F	K	F	K	F	K	F	K	F	K
1	0	3	0	3	0	6	0	3	0	2	0	3
2	0	3	0	3	0	6	0	3	0	2	0	3
3	0	3	0	3	0	6	0	3	0	2	0	3
4	0	3	0	3	0	6	0	3	0	2	0	3

Table 13. Owner information system error

Trial	T1		T2		T3		T4		T5		T6	
	F	K	F	K	F	K	F	K	F	K	F	K
1	0	3	0	3	0	4	1	4	0	3	0	4
2	0	3	0	3	0	4	0	4	0	3	0	4
3	0	3	0	3	0	4	0	4	0	3	0	4
4	0	3	0	3	0	4	0	4	0	3	0	4

Satisfaction is used to measure user satisfaction with the developed system through a questionnaire. The measurement is conducted using the System Usability Scale (SUS), which employs a Likert Scale ranging from 1 (strongly disagree) to 5 (strongly agree). The questionnaire was created using Google Forms and consisted of 10 statements related to the overall use of the XYZ boarding house information system (Table 14). The statements were developed based on the System Usability Scale instrument by [21]. The responses collected via Google Forms were then processed according to the SUS scoring guidelines. Using these guidelines, the total evaluation results were obtained as presented in Table 15.

Table 14. Satisfaction questionnaire results from Google form

Respon-dent	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Tenant 1	4	1	5	1	5	1	5	1	4	1
Tenant 2	4	2	4	2	5	2	4	2	5	2
Tenant 3	5	1	5	1	5	1	5	1	4	1
Tenant 4	4	1	5	1	4	1	5	1	5	1
Owner	5	2	5	1	5	1	5	2	5	1

Table 15. Questionnaire results based on SUS

Respondent	Total Statements		Score	Total
	Odd	Even		(Score x 2,5)
Tenant 1	18	20	38	95
Tenant 2	17	15	32	80
Tenant 3	19	20	39	97.5
Tenant 4	18	20	38	95
Owner	20	18	38	95
Total Assessment				367.5

The assessment result using the SUS yielded a total score of 367.5. This result was then processed by calculating the average score using the standard formula. Based on the calculation results, the average score obtained was 73.5. Referring to the SUS evaluation by [22], a SUS score of 73.5 is classified as a grade of B-. This result indicates that the developed information system falls within the "good" category and is considered marginally acceptable [23]. Therefore, the design of the information system for the XYZ boarding house has been successfully implemented and functions as intended. The respondents also indicated that the information system was well-received.

4. Conclusion

Based on the research findings and the development of the Android-based financial management information system for the XYZ boarding house, the system has proven effective in facilitating the transition from manual to digital payment processing. By accommodating room-specific rental prices and due dates, the system enhances monitoring capabilities for the boarding house owner and simplifies payment processes for tenants. This improvement is critical in addressing the operational inefficiencies previously caused by inconsistent record-keeping and delayed payments.

The results of the usability testing conducted with end users—including both the boarding house owner and tenants—demonstrated strong performance across all five evaluated dimensions: learnability, efficiency, memorability, error rate, and user satisfaction. The tenant system achieved a learnability rate of 98% and an efficiency score of 0.10 goals/sec, while the owner system recorded a perfect learnability score of 100% and an efficiency score of 0.04 goals/sec. Furthermore, the memorability results revealed improved task completion times on the seventh day of testing, indicating intuitive system design. The recorded error rates for both systems were below the acceptable threshold of 0.7, while the average SUS (System Usability Scale) score of 73.5 placed the system in the “good” category, suggesting strong user acceptance.

Despite these positive outcomes, several enhancements are recommended to increase system robustness and scalability. First, implementing a two-factor authentication mechanism using email or SMS verification could significantly strengthen user account security. Second, refining the user interface (UI) and user experience (UX) design would increase engagement and usability, especially for first-time users. Third, integrating a secure and automated payment gateway would allow real-time transaction processing and reconciliation, further improving operational efficiency.

Lastly, the system architecture could be generalized to serve a broader user base, enabling its adoption by other boarding house operators with similar management needs.

Future research could focus on evaluating the system's scalability when deployed in multiple boarding houses with varying operational structures. Additionally, longitudinal studies could be conducted to assess user engagement and system performance over an extended period. Incorporating advanced analytics and reporting features may also add strategic value by providing insights into payment trends, tenant behavior, and occupancy rates. These enhancements would further align the system with modern smart property management practices, supporting digital transformation in the housing sector.

Acknowledgements

This work has been supported by ELLIIT, the Swedish Strategic Research Area in IT and Mobile Communications.

References:

[1]. Kalinowska, K., Wojnowski, W., & Tobiszewski, M. (2021). Smartphones as tools for equitable food quality assessment. *Trends in Food Science & Technology*, *111*, 271-279.

[2]. Biswas, T. R., Hossain, M. Z., & Comite, U. (2024). Role of Management Information Systems in Enhancing Decision-Making in Large-Scale Organizations. *Pacific Journal of Business Innovation and Strategy*, *1*(1), 5-18.

[3]. Fernando, W. M., et al. (2025). Storm in a teacup: implications of mobile phone literacy on sustainable smallholder agri-food supply chains in developing economies. *The International Journal of Logistics Management*, *36*(2), 452-473.

[4]. Eusufzai, F., et al. (2024). Personal internet of things networks: An overview of 3GPP architecture, applications, key technologies, and future trends. *International journal of intelligent networks*, *5*, 77-91.

[5]. Hossain, M. I. (2023). Software development life cycle (SDLC) methodologies for information systems project management. *International Journal for Multidisciplinary Research*, *5*(5), 1-36.

[6]. Yang, C. H., et al. (2023). Evaluation of smart long-term care information strategy portfolio decision model: the national healthcare environment in Taiwan. *Annals of Operations Research*, *326*(1), 505-536.

[7]. Emadi, J. (2023). The Development of a Design Theory for Web Based Information Systems. *Journal of Robotics Spectrum*, *1*, 013-023.

[8]. Farshidi, S., Kwantes, I. B., & Jansen, S. (2024). Business process modeling language selection for research modelers. *Software and Systems Modeling*, *23*(1), 137-162.

[9]. Ouariach, S., Ouariach, F. Z., & Khaldi, M. (2025). A software engineering approach for conceptualising an online learning scenario for a deductive approach. *International Journal of Intelligent Engineering Informatics*, *13*(1), 1-25.

[10]. Elizarov, R., et al. (2021). Kotlin coroutines: design and implementation. *Proceedings of the 2021 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*, 68-84.

[11]. Lestari, B., Rifiani, P. I., & Gati, A. B. (2021). The use of the usability scale system as an evaluation of the kampung heritage kajoetangan guide ebook application. *European Journal of Business and Management Research*, *6*(6), 156-161.

[12]. Ahmad, N. A. N., Hamid, N. I. M., & Lokman, A. M. (2021). Performing Usability Evaluation on Multi-Platform Based Application for Efficiency, Effectiveness and Satisfaction Enhancement. *International Journal of Interactive Mobile Technologies*, *15*(10).

[13]. Geirhos, R., et al. (2021). Partial success in closing the gap between human and machine vision. *Advances in Neural Information Processing Systems*, *34*, 23885-23899.

[14]. Alrazgan, M., & AlRajhi, N. A. (2022). *Learnability evaluation model for Android mobile applications*. *International Journal of Computer Applications*, *184*(35).

[15]. Thillaieaswaran, B., & Pasupathy, S. (2021). Learnability Metric in Software Quality Assurance. *Annals of the Romanian Society for Cell Biology*, *25*(2), 3993-3997.

[16]. Najihi, S., et al. (2022). Software Testing from an Agile and Traditional view. *Procedia Computer Science*, *203*, 775-782.

[17]. Sekgweleo, T., & Iyamu, T. (2022). Understanding the factors that influence software testing through moments of translation. *Journal of Systems and Information Technology*, *24*(3), 202-220.

[18]. Giray, G. (2021). An assessment of student satisfaction with e-learning: An empirical study with computer and software engineering undergraduate students in Turkey under pandemic conditions. *Education and information technologies*, *26*(6), 6651-6673.

[19]. Jayagopal, D., Lubin, J., & Chasins, S. E. (2022). Exploring the learnability of program synthesizers by novice programmers. *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, 1-15.

[20]. Kim, Y.-M., & Park, M.-H. (2024). *Main causes of missing errors during software testing*. *Journal of the Korea Society of Computer and Information*, *29*(6), 89–100.

[21]. Vlachogianni, P., & Tselios, N. (2022). Perceived usability evaluation of educational technology using the System Usability Scale (SUS): A systematic review. *Journal of Research on Technology in Education*, *54*(3), 392-409. Doi: 10.1080/15391523.2020.1867938

[22]. Baumgartner, J., et al. (2021). Questionnaire experience and the hybrid System Usability Scale: Using a novel concept to evaluate a new instrument. *International Journal of Human-Computer Studies*, *147*, 102575.

[23]. Kortum, P., Acemyan, C. Z., & Oswald, F. L. (2021). Is it time to go positive? Assessing the positively worded system usability scale (SUS). *Human factors*, *63*(6), 987-998.