JISA (Jurnal Informatika dan Sains)
Vol. 07, No. 02, December 2024
e-ISSN: 2614-8404
p-ISSN:2776-3234

Development of Paramadina Roomhub Application As Room Booking System Using Waterfall Method

Reza Arif Maulana[1*)], Muhamad Adillah Fatih[2], Lintang Arbi Suto[3], Muhammad Darwis[4]
[1234]Program Studi Teknik Informatika, Fakultas Ilmu Rekayasa, Universitas Paramadina
Email: [1]reza.maulana@students.paramadina.ac.id, [2]muhamad.fatih@students.paramadina.ac.id,
[3]lintang.suto@students.paramadina.ac.id, [4]muhammad.darwis@paramadina.ac.id

Abstract – The manual room booking process at Paramadina University faces several challenges, including inefficiencies, human errors, and communication breakdowns between applicants and facility managers. These issues often result in room booking conflicts and delays, hindering effective facility usage. The need for a more streamlined, user-friendly solution led to the development of the Roomhub Paramadina application, a web-based system designed to facilitate room bookings within the university. The objective of this research is to design and implement a digital room booking system that simplifies the current manual process. The application was developed using the Software Development Life Cycle (SDLC) with the Waterfall framework. This approach ensures structured and systematic development, enabling the team to follow sequential stages from requirements gathering to testing. The Roomhub application allows users to view available rooms, submit booking requests, and receive confirmations directly via an online platform. The system integrates with existing university infrastructure and ensures real-time room availability, reducing errors such as double bookings. Black-box testing was employed to verify the application's functionality, and the results confirmed that the system meets user needs while significantly improving booking efficiency. By automating room bookings and enhancing communication between the parties involved, this research offers a practical solution that improves operational efficiency, reduces paperwork, and minimizes scheduling conflicts, thereby benefiting the entire academic community at Paramadina University.

Keywords – Room Booking System, Web-Based Application, Software Development, Waterfall Methodology, Blackbox Testing

I. INTRODUCTION

Paramadina University is located in Jakarta, Indonesia, and was established in 1998[1]. Currently, Paramadina University has three campuses distributed across the Jabodetabek area, with one located in Cipayung, East Jakarta. The university offers a range of public facilities, including classrooms, laboratories, lecture halls, auditoriums, meeting rooms, sports fields, and other communal areas[2].

In providing its services, Paramadina University permits the Paramadina academic community to utilize these facilities for purposes other than their primary function, namely as lecture halls. The aforementioned facilities are available to all Paramadina academicians who have been granted the requisite access rights, which include students, lecturers, and university employees. To gain permission to use the facilities, individuals must submit a request for a facility booking.

Paramadina University has established a dedicated unit to oversee the utilization of public facilities within its campus. This unit is responsible for maintaining accurate records, organizing scheduling, and issuing permits for each facility. Furthermore, it provides supervisory support to ensure the safe and effective operation of these facilities. Paramadina University has established a dedicated public facility management team for each of its campuses. The Study Program Service Unit (UPPS) oversees the management of public facilities at the Cipayung Campus. Consequently, any Paramadina academic seeking to utilize public facilities on this campus must liaise with or submit a request to the relevant team via email.

To utilize campus facilities, specifically rooms, whether regular classrooms, conference rooms, lecture halls, or other such rooms, the Paramadina academic community must submit a book request to UPPS via email. Subsequently, the UPPS will process the request by forwarding it to the Directorate of Student Affairs and Business Incubator (DKI) to facilitate a follow-up response. If the request is approved, the DKI will issue a formal approval email to the UPPS, which will then process and record the request and provide relevant details to the Public Facilities Unit and the applicant. However, if the request is not approved, the DKI will issue an official rejection email, which will be subsequently conveyed to the applicant.

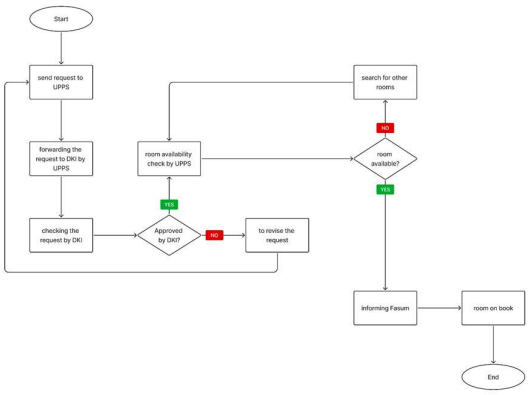

Figure 1. Room Booking Flow

It should be noted that the room identified in the booking application may not be suitable for use, despite the approval of DKI. There are several potential causes for this, such as maintenance work, the room being occupied, or it having been booked to another party. In

e-ISSN: 2614-8404
p-ISSN:2776-3234

such circumstances, the applicant is expected to identify an alternative room and submit a revised application, replacing the initial room specified.

Furthermore, additional factors influence the process of borrowing facilities, particularly rooms. These include the regulations that accompany the borrowing process. Based on the author's and his colleagues' direct experience of applying for a room booking, the regulations include:

1. Requests for room use must be made at least seven days before the date in question.
2. Should the room in question be used for an event, the event proposal must be approved in advance.
3. All student activities must be approved by the DKI.
4. Requests for room use must be accompanied by details of productive activities.

With such a process, and influenced by the rules in the booking process, the author considers the process inefficient. The application must pass through many stages involving many parties, so the bureaucracy is complex and time-consuming.

The conventional and manual facility booking systems and processes are susceptible to human error. There is a high probability of data loss or inadequate documentation, and communication breakdowns between applicants and room managers, which can result in erroneous information regarding space availability, double recording, potential clashes in space usage schedules, and coordination challenges[3], [4], [5]. The precise specifications, facilities/infrastructure, and status of each room are not clearly defined, making it challenging for applicants to select the optimal room[6].

With technological advances, the problems that exist in the conventional process can be avoided. Researchers take advantage of this moment to develop innovations that facilitate the process of booking facilities, especially rooms. Based on this momentum, the author conducted research with the objective of developing a web-based application designed to facilitate the management of public facilities, including the booking process. This application is designed to markedly enhance the velocity, convenience, and efficacy of public facility services, particularly in regard to rooms. The application was developed using the Software Development Life Cycle (SDLC) approach with the Waterfall Framework. The Waterfall Framework was selected for its emphasis on structured and sequential steps, which result in comprehensive and detailed documentation[7], [8]. The sequential nature of the stages facilitates comprehension of the process by team members from both technical and non-technical perspectives[9]. Allows for better project control and facilitates communication within the team.

II. RESEARCH METHODS

This research employs descriptive qualitative methods to gain a comprehensive understanding of the phenomenon through detailed descriptions, making it a suitable method for application development. In this research, application development uses the SDLC (Software Development Life Cycle) approach with a Waterfall framework. In the context of application development, the SDLC was selected as the preferred methodology due to its capacity to provide a structured framework for the effective planning, development, and management of applications. The SDLC is an economical and commonly used methodology for developing quality software. It comprises several models that facilitate a simple and systematic development process[10].

The Waterfall framework is frequently selected for its linear and sequential nature, wherein each phase must be completed before the next phase begins. The Waterfall framework is frequently selected for its linear and sequential nature, wherein each phase must be completed before the subsequent phase commences. This structured approach is well-suited to projects with clearly defined and unchanging requirements, while also reducing the likelihood of significant alterations during the development phase[11].

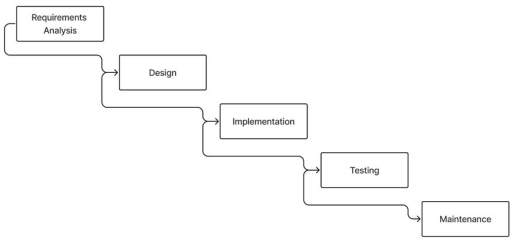

Figure 2. Waterfall Framework

The stages of the Waterfall model are as follows[12], [13]:

1. Requirement Analysis: this stage is the initial stage that serves to collect information about user needs and application requirements to be made.
2. Design: at this stage, an overall application design is made, starting from application architecture, database architecture, and user interface design.
3. Implementation: This stage involves creating an application based on the design that has been made in the previous stage.
4. Testing: at this stage, the application that has been created is then tested to ensure that the application runs well and in accordance with the application requirements.
5. Maintenance: after the application has been implemented and tested, further maintenance of the application is carried out which includes bug fixes, improving application performance, optimization, and adding new features.

e-ISSN: 2614-8404
p-ISSN:2776-3234

The selection of the Waterfall method over other SDLC frameworks, such as Agile or Spiral, is based on the necessity for a more structured and controlled approach to application development projects with a fixed scope. While alternative frameworks, such as Agile, are more flexible and accommodate iterative changes, the Waterfall framework provides greater stability in projects with clearly defined requirements from the outset[11].

A. Requirement Analysis

At this stage, the author collects information related to user needs and applications that will be designed and developed. This process involved interviews with the academic community of Paramadina University, including lecturers, students, and managers of campus public facilities. The interviews aimed to gain an in-depth understanding of user needs. These interviews are important in analyzing the needs of application development because they help identify features that meet user needs[14].

The objective of the interviews was to gain insight into the process of applying for a room booking. The interviews were conducted either in person or via an online platform, depending on the respondent's readiness and preference. The duration of the interviews varied, with an average time of approximately 15 to 30 minutes per session. The data obtained from the interviews were meticulously recorded and analyzed to identify common patterns related to user and system needs. The results of this analysis serve as the foundation for formulating appropriate system specifications that meet those needs.

B. Design

System design is carried out using the Unified Modelling Language (UML) modeling concept which includes structural and behavioral aspects, through use case diagrams, activity diagrams, and class diagrams. These three diagrams represent a comprehensive modeling of the system architecture of the RoomHub Paramadina application. UML is a visual modeling language used to describe the needs of software analysis and design[15]. UML has the potential to serve as a detailed and comprehensive blueprint. In this context, UML can provide detailed information about the source code, facilitating program understanding, debugging, and maintenance, and enabling the reading of program code and its conversion into diagrammatic form[16].

C. Implementation

The implementation phase entails the creation of an application that adheres to the specifications outlined in the system design. The chosen framework comprises Next.js for the client side, Express.js for the server side, Node.js as the runtime, and MongoDB as the database.

a. Next.js

Next.js is a React-based framework that enables the development of web applications with a range of features, including integrated CSS, built-in routing, layout, and image and font optimization. Next.js permits the utilization of pre-rendering and server-side rendering techniques to reduce the time required for application loading. Next.js provides efficient development support by optimizing application performance through code splitting, prefetching, and lazy loading. Additionally, Next.js offers comprehensive documentation and interactive courses to facilitate the acquisition of knowledge and proficiency in the proper use of Next.js[17].

b. Express.js

Express.js is a server-side framework for Node.js that has gained considerable popularity due to its efficacy in facilitating the development of web applications and Application Programming Interfaces (APIs). Express.js was designed to facilitate the construction of Node. js-based applications. It offers a suite of robust yet straightforward tools for the development of diverse web applications and API services. The framework is renowned for its capacity to process Hypertext Transfer Protocol (HTTP) requests, route data, utilize middleware, manage requests and responses, and provide features that expedite the development process. Additionally, Express.js enables the integration of middleware and other Node.js modules, considerably expanding its functional capabilities[18], [19].

c. Node.js

Node.js is an open-source runtime environment and cross-platform for executing JavaScript code. Node.js, released in 2009 by Ryan Dahl, enables developers to create client-side and server-side applications using JavaScript. Node.js runs using the V8 JavaScript engine, thereby enabling the execution of JavaScript code outside the browser environment. Node.js serves as the foundation for Express.js and Next.js. Express.js employs Node.js as a runtime environment for developing web applications with JavaScript. Next.js, at a more advanced level, utilizes Node.js and, on occasion, Express.js as a basis to provide a more robust set of tools for building more complex and feature-rich web applications[20], [21].

d. MongoDB

MongoDB is a document-based NoSQL database that stores data in JavaScript Object Notation (JSON) or Binary JSON (BSON) format. In contrast to relational databases such as MySQL or PostgreSQL, MongoDB is a non-relational database that permits more flexible and scalable data storage, thereby rendering MongoDB a reliable choice for storing large volumes of data[22], [23].

D. Testing

e-ISSN: 2614-8404
p-ISSN:2776-3234

At this stage, the system is tested using the Blackbox method to ensure that the application functions properly and meets user needs. The Blackbox testing method is an approach that tests an application without considering the internal structure and logic of the code. This approach focuses on analyzing the input and output of the application, without requiring a deep understanding of the source code or application design[24].

A number of references from various scientific journals indicate that this test is oriented toward user needs and ensures application conformance to predefined specifications. Furthermore, it is carried out without access to internal system details. For example, research conducted by Dwi et al. in the article "Testing Learning Media for English Learning Applications Using Black-Box Testing Based on Equivalence Partitions" [25] demonstrates the efficacy of black-box testing in identifying defects in English learning media applications. Other references also illustrate that black-box testing is an effective approach for detecting errors without requiring access to the internal workings or source code of an application[26], [27], [28], [29].

E. Maintenance

Although the maintenance stage represents the final phase of the Waterfall method, this research concludes at the testing stage, as the objectives of the design process for this application have been met and the issue has been resolved.

III. RESULT AND DISCUSSION

A. Requirement Analysis

The author identifies the scope of applicability based on information obtained through interviews with relevant parties. The scope of applicability is constrained by the data provided by the facility manager, which pertains to the rooms available on campus and the booking periods permitted in accordance with campus policy. This data is essential for the author to develop a room booking application.

B. Design

System design is carried out using the Unified Modelling Language (UML) modeling concept which includes structural and behavioral structures with use case diagrams, activity diagrams, and class diagrams, where the three diagrams already reflect the system architecture modeling of the RoomHub Paramadina application.

a. Use Case Diagram

A use case diagram provides an overview of the overall application, including the interactions between users (actors) and the RoomHub Paramadina application (use cases). A use case diagram comprises four principal components: actors, use cases, use case diagrams, and use case diagram scenarios. In this diagram, the author delineates the use case for each scenario and determines the interaction between the actor and the system use case. The diagram contains two cases: the Room Information case and the Room Loan use case. The design diagram that elucidates the interaction between actors and use cases is illustrated in Figure 3.

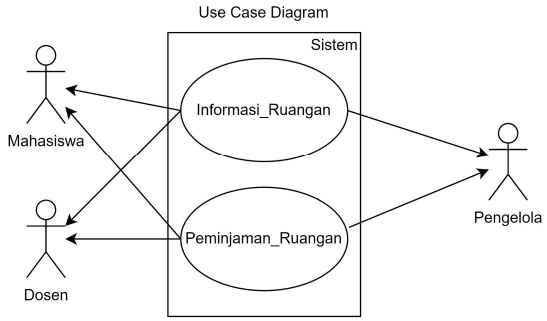

Figure 3. Use Case Diagram

b. Activity Diagram

The next diagram is the Activity Diagram. An activity diagram is a flexible instrument to describe the behavior or workings of a system and the internal logic of complex operations. This diagram helps describe in a simple way how the flow of events from the use case works. There are three Activity Diagrams shown in Figures 4, 5, and 6 to explain the behavior or workings of each use case.

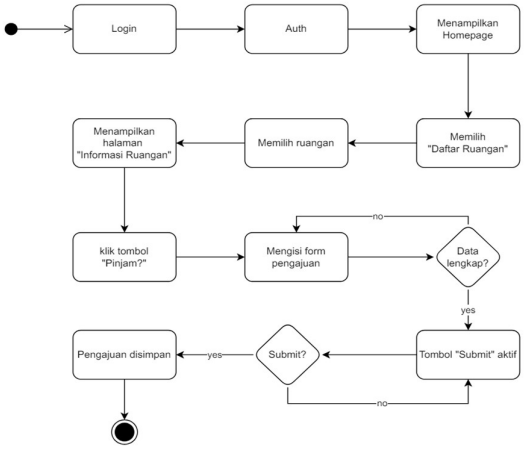

Figure 4. Activity Diagram of Room Booking

e-ISSN: 2614-8404
p-ISSN:2776-3234

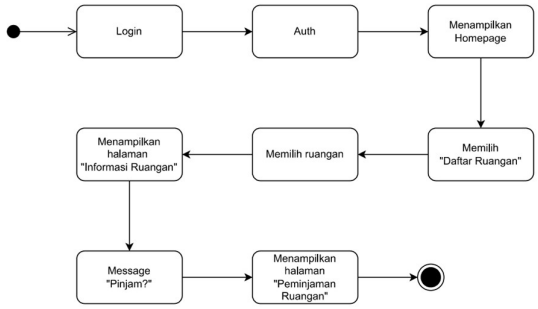

Figure 5. Activity Diagram of Room Information for Lecturers and Students

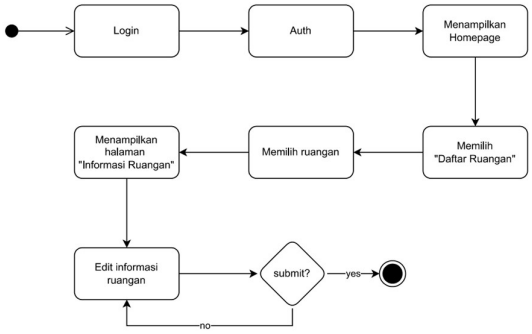

Figure 6. Activity Diagram of Room Information for Manager

c. Diagram Class

A class diagram is a UML diagram that displays the classes in software and the relationships between them[3]. The class diagram designed by the author comprises a User class that delegates its attributes and methods to the Student, Lecturer, and Manager classes, which possess a single distinctive method. Additionally, there is a Booking class and a Room class. Each class is also linked by a relation to delineate the association between classes. The class diagram for the RoomHub Paramadina application is illustrated in Figure 7.

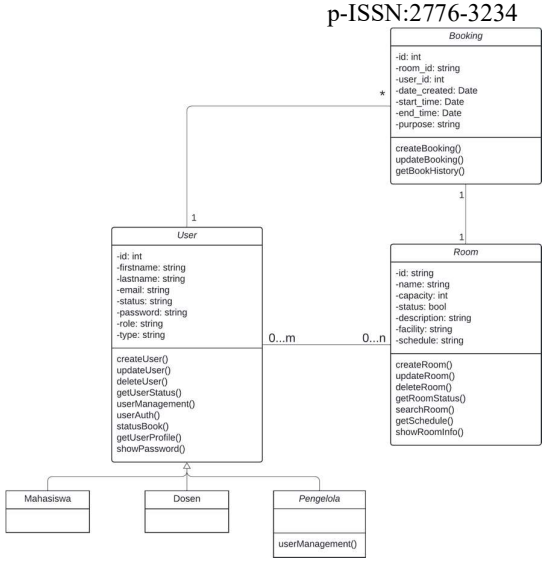

Figure 7. Class Diagram of RoomHub Paramadina

C. Implementation

The results of designing an application using the waterfall method on the RoomHub Paramadina web-based application, which produces several menus that will be utilized by the user, are presented herein. To gain access, users must first click the "Sign in with Google" button, which will prompt them to enter their Paramadina University student Google account credentials. Once the login process is complete, users are granted access to the feature that allows them to view room information and make room reservations. Once a room has been reserved, the user is able to view the booking history. Subsequently, the manager is able to modify room information, add new rooms, and delete existing rooms.

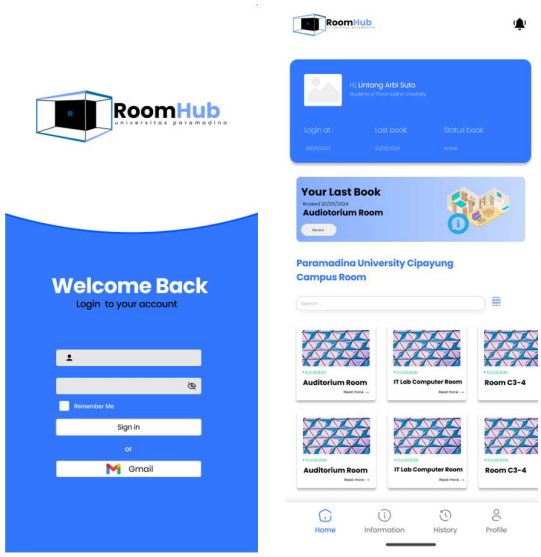

Figure 8 & 9. Login Page and Landing Page

e-ISSN: 2614-8404
p-ISSN:2776-3234

Figure 8 illustrates the login page. This page enables users to input the requisite account information, including the username and password. Additionally, the interface provides a button for logging in with a Google account, as well as options for password recollection and viewing. Moreover, upon successful authentication, the user is redirected to the landing page depicted in Figure 9. This page presents the user with information regarding their account, including login status, the most recent booking, and the status of any active bookings. Additionally, a list of available rooms on campus is provided, with the option of searching for specific criteria and viewing the status of room availability. The navigation bar located at the bottom of the page provides users with convenient access to the Home, Information, History, and Profile pages.

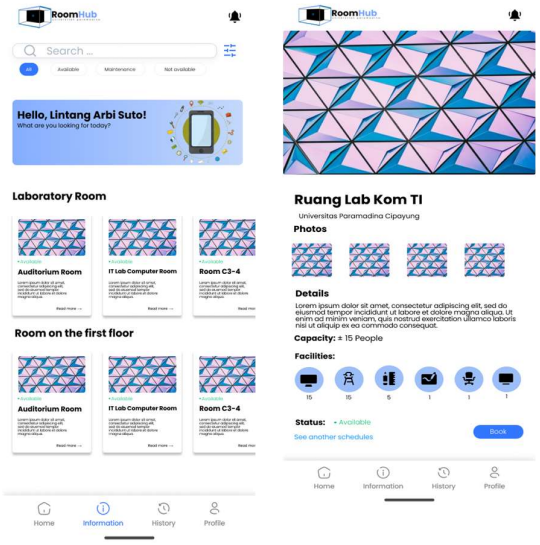

Figure 10 & 11. Room Information and Room Details page

Next one is the Room Information page, as illustrated in Figure 10, which permits users to search and filter rooms based on their availability (All, Available, Unavailable). A list of rooms is provided, with each room designated by a unique availability status, such as Room C3-4. Upon selecting a room card, the user will be redirected to the comprehensive room information displayed in Figure 11. The user is able to view photographs of the room, a succinct description, the room's capacity, the facilities that are available, and the room's current status in terms of availability. A button labeled "Book" is provided for users to reserve the room. Additionally, a navigation bar is located at the bottom of the page, allowing users to navigate between different application pages.

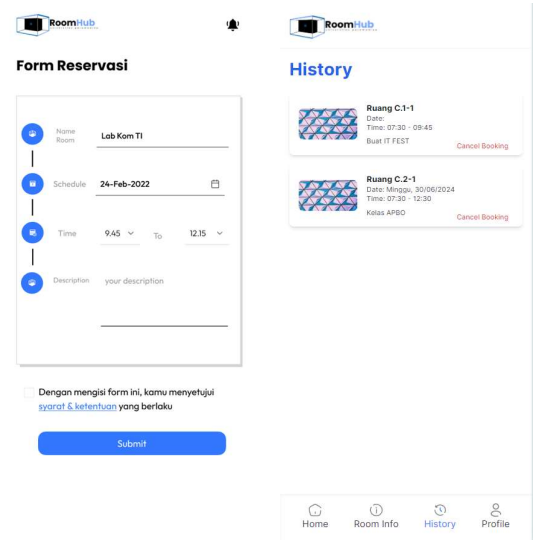

Figure 12 & 13. Room Booking Form and Room Booking History Page

Subsequently, upon clicking the "Book" button, the user will be redirected to the Booking Form page, as illustrated in Figure 12. This page allows the user to input the name of the room, the date, time, and a description of the booking. Additionally, the user must agree to the terms and conditions before proceeding. Subsequently, the user will be redirected to the History page, as illustrated in Figure 13. The History page presents a chronological record of room bookings, accompanied by pertinent details such as the time, event, and the option to cancel the reservation.

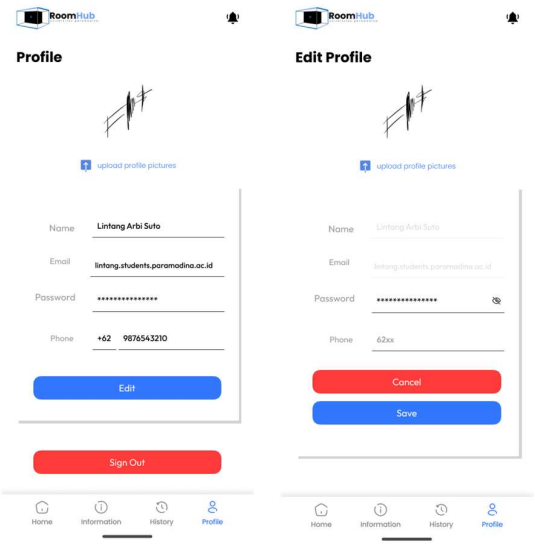

Figure 14 & 15. User Profile and Edit Profile page

Additionally, the user is able to view the account profile by navigating to the User Profile page, as illustrated in Figure 14. This section contains the user

e-ISSN: 2614-8404
p-ISSN:2776-3234

profile details, including the user's name, email address, and other personal information. It also includes an option to edit the profile data, should the user wish to do so. Moreover, should the user wish to modify the profile, an Edit button will direct the user to the Edit Profile page, as illustrated in Figure 15. The user is afforded the opportunity to modify the password and to add or alter the phone number. Once the requisite modifications have been made, the user may save the profile data by selecting the Save button.

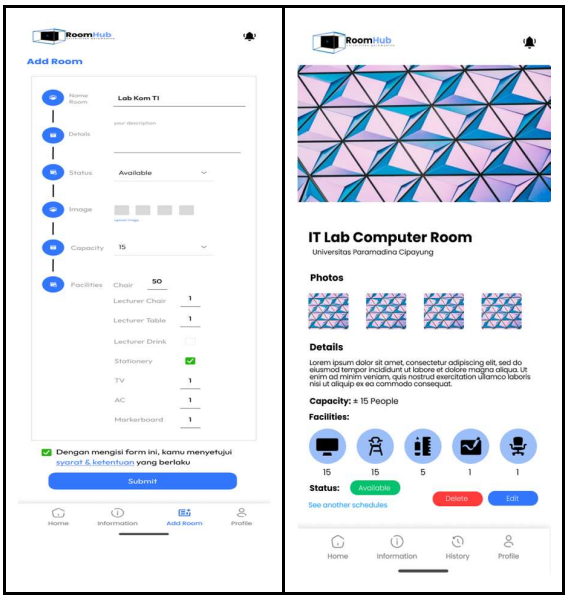

Figure 16 & 17. Room Addition and Room Information Page for Manager

Subsequently, one should proceed to the administrator or manager section, where the administrator is able to create a new room. Once the manager has successfully completed the login process, he or she will be redirected to the Admin Dashboard page. Subsequently, the manager is able to navigate to the button labelled "Add Room" in order to add a room. Figure 16 illustrates the Room Addition Page, which allows the manager to input the necessary information to add a new room to the system. This includes details such as the room name, capacity, and available facilities. To finalize the addition of the room, the manager must press the "Submit" button. Should the manager wish to modify the data pertaining to the room in question, they may navigate to the landing page, which lists all rooms. Subsequently, the manager may select a room for which they wish to modify the data, which will direct them to the Room Information page for Managers, as illustrated in Figure 17. This page enables the manager to view comprehensive details regarding the room, including capacity, facilities, and availability status, with the option to edit or manage the room information.

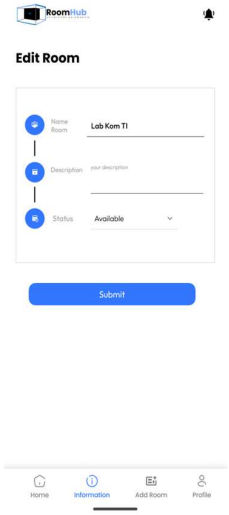

Figure 18. Room Information Edit Page

To modify room data, the manager must first press the "Edit" button. This action will direct the manager to the "Edit Room" page, as illustrated in Figure 18. This page enables the manager to modify room-related data, including the room name, capacity, facilities, and current availability status, which have been previously documented within the system.

D. Testing

At this stage, the author assesses the functionality of the proposed application. The objective of testing is to ascertain that the application functions in accordance with the specifications set forth in the design. The Black-box method is employed for the purpose of testing. The results of the tests are presented in tabular form.

Table 1Test Result Using Blackbox Testing Method

Test Case	**Condition**	**Instruction**	**Expected Result**	**Test Result**	**Explanation**
Login	Email and password	Enter a valid email and password, then click “Login”.	User successfully logs in and is	Login succeeds	The system validates the user's credentials,

e-ISSN: 2614-8404
p-ISSN:2776-3234

	input are correct		redirected to the main page		redirecting to the dashboard if valid.
Login	Email and password input are incorrect	Enter the wrong email or password	Displays the error message "Email or password is incorrect"	Error message displayed	The system checks the credentials, denies access and displays an error message.
Room Booking	Input of available rooms	Select an available room, fill in the data, click "Book"	The room is successfully booked and appears in the booking list	Booking successful	The system saves the booking data and updates the user's booking list.
Room Information	Display detailed room information	Select a room to see detailed information	Detail room information displayed (facilities, capacity, etc.)	Information shown	The system retrieves the room data from the database and displays it to the user.
Room Information Edit (Admin)	Admin edits the room information	Admin changes the room information, then clicks "Save"	Room information updated successfully	Information updated successfully	The system saves the changes to the room data in the database.
Delete Room (Admin)	Admin deletes a room	Admin select a room and click "Delete"	The room is successfully removed from the system	Room removed	The system deletes the room data from the database and updates the room list.
Room Search	Room search based on keywords	Enter the room name or room code in the search field	Rooms that match the keywords appear	The room appears	The system searches by keyword and displays relevant results.

e-ISSN: 2614-8404
p-ISSN:2776-3234

IV. CONCLUSION

The findings of the research conducted thus far indicate that the web-based RoomHub Paramadina application architecture with a mobile display has been successfully developed. The application offers the ability to view room information and make reservations for rooms at Paramadina University, particularly at the Cipayung Campus, without the necessity of direct interaction or submission of a loan letter to the Paramadina Student Council and Business Incubator (DKI) and Cipayung Campus academic services. The RoomHub Paramadina architecture has been equipped with a draft interface that can be tested at an early stage in accordance with the concept of prototyping, which allows for rapid testing to obtain an initial overview of the application. The prototyping process has only been completed once, with subsequent cycles planned to incorporate improvements. At this preliminary stage, the interface design and implementation have yielded the anticipated outcomes, yet further advancement is necessary. The Unified Modelling Language (UML) was employed as an appropriate notation to model the system architecture and provide a transparent representation of the system's operational logic.

The following recommendations are offered for consideration: System design using the object-oriented analysis and design (OOAD) method requires meticulous planning, particularly in object-based design, to ensure that the system model document is comprehensive and suitable for immediate implementation. Moreover, software development based on the prototyping paradigm necessitates the completion of several additional cycles to ensure that the resulting software is optimized to meet user needs and analysis.

REFERENCES

[1] Universitas Paramadina, "Tentang Universitas Paramadina." Accessed: Oct. 24, 2024. [Online]. Available: https://paramadina.ac.id/tentang-universitas-paramadina/

[2] Universitas Paramadina, "Universitas Paramadina Kampus Cipayung." Accessed: Oct. 24, 2024. [Online]. Available: https://paramadina.ac.id/kampus-cipayung/

[3] H. Singh and R. R. Shah, "BOOKiiIT - Designing a Venue Booking System (Technical Demo)," in *2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM)*, IEEE, Sep. 2020, pp. 287–291. doi: 10.1109/BigMM50055.2020.00050.

[4] Sabar Rudiarto, Saruni Dwiasnati, and Agung Susilo Yuda Irawan, "Design and build expert system applications using forward chaining methods to manage web-based space management," *Global Journal of Engineering and Technology Advances*, vol. 10, no. 2, pp. 009–017, Feb. 2022, doi: 10.30574/gjeta.2022.10.2.0115.

[5] F. A. Pratama, S. P. D. Kristiana, and C. M. Lukito, "The Implementation of SST as the System of Room Reservation for Students' Organization at Atma Jaya BSD Campus," *SISFORMA*, vol. 8, no. 2, pp. 44–50, Jan. 2022, doi: 10.24167/sisforma.v8i2.3434.

[6] T. Bai, Y. Tian, and W. Li, "Research on Shared Information Management in University College Reference Room," 2018, pp. 361–371. doi: 10.1007/978-3-319-93719-9_49.

[7] K. Wiguna and D. Mahdiana, "Analysis of Information Systems Development Methods: A Literature Review," *INOVTEK Polbeng - Seri Informatika*, vol. 8, no. 2, p. 472, Nov. 2023, doi: 10.35314/isi.v8i2.3753.

[8] S. Pargaonkar, "A Comprehensive Research Analysis of Software Development Life Cycle (SDLC) Agile & Waterfall Model Advantages, Disadvantages, and Application Suitability in Software Quality Engineering," *International Journal of Scientific and Research Publications*, vol. 13, no. 8, pp. 120–124, Aug. 2023, doi: 10.29322/IJSRP.13.08.2023.p14015.

[9] M. I. H. -, "Software Development Life Cycle (SDLC) Methodologies for Information Systems Project Management," *International Journal For Multidisciplinary Research*, vol. 5, no. 5, Sep. 2023, doi: 10.36948/ijfmr.2023.v05i05.6223.

[10] A. Gupta, "Comparative Study of Different SDLC Models," *Int J Res Appl Sci Eng Technol*, vol. 9, no. 11, pp. 73–80, Nov. 2021, doi: 10.22214/ijraset.2021.38736.

[11] N. Rachma and I. Muhlas, "Comparison Of Waterfall And Prototyping Models In Research And Development (R&D) Methods For Android-Based Learning Application Design," *Jurnal Inovatif : Inovasi Teknologi Informasi dan Informatika*, vol. 5, no. 1, p. 36, Aug. 2022, doi: 10.32832/inova-tif.v5i1.7927.

[12] R. S. Ghumatkar and A. Date, "Software Development Life Cycle (SDLC)," *Int J Res Appl Sci Eng Technol*, vol. 11, no. 11, pp. 1162–1165, Nov. 2023, doi: 10.22214/ijraset.2023.56554.

[13] H. J. Christanto and Y. A. Singgalen, "Analysis and Design of Student Guidance Information System through Software Development Life Cycle (SDLC) dan Waterfall Model," *Journal of Information Systems and Informatics*, vol. 5, no. 1, pp. 259–270, Mar. 2023, doi: 10.51519/journalisi.v5i1.443.

[14] Ștefan-D. GEORGESCU and I.-E. ANASTASIU, "THE INTERVIEW AS A QUALITATIVE RESEARCH INSTRUMENT," Feb. 2022. doi: 10.24818/IMC/2021/05.14.

e-ISSN: 2614-8404
p-ISSN:2776-3234

[15] D. Bork, D. Karagiannis, and B. Pittl, "A survey of modeling language specification techniques," *Inf Syst*, vol. 87, p. 101425, Jan. 2020, doi: 10.1016/j.is.2019.101425.

[16] D. Bork, D. Karagiannis, and B. Pittl, "A survey of modeling language specification techniques," *Inf Syst*, vol. 87, p. 101425, Jan. 2020, doi: 10.1016/j.is.2019.101425.

[17] Inc. Vercel, "Introduction | Next.js." Accessed: Oct. 24, 2024. [Online]. Available: https://nextjs.org/docs

[18] OpenJS Foundation, "Express," 2024, Accessed: Oct. 24, 2024. [Online]. Available: https://expressjs.com/

[19] PROF. B. V. Karlatthe, "MERN Stack Based User Authentication Technique for Evernote Application," *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, vol. 07, no. 11, pp. 1–11, Nov. 2023, doi: 10.55041/IJSREM27053.

[20] OpenJS Foundation, "Introduction to Node.js." Accessed: Oct. 24, 2024. [Online]. Available: https://nodejs.org/en/learn/getting-started/introduction-to-nodejs

[21] Dicoding, "Tutorial Node.js: Apa itu Node.js dan dasar-dasar Node.js." Accessed: Oct. 24, 2024. [Online]. Available: https://www.dicoding.com/blog/apa-itu-node-js/

[22] Inc. MongoDB, "What Is MongoDB." Accessed: Oct. 24, 2024. [Online]. Available: https://www.mongodb.com/company/what-is-mongodb

[23] P. S. W. Ei, "Learning in horizontal scaling MongoDB data sharding on windows," *i-manager's Journal on Software Engineering*, vol. 17, no. 3, p. 1, 2023, doi: 10.26634/jse.17.3.19790.

[24] Gilang Ryan Fernandes and Ika Mei Lina, "Boundary Value Analysis Testing Against Library Applications Using the Black Box Method as System Performance Optimization," *Jurnal E-Komtek (Elektro-Komputer-Teknik)*, vol. 5, no. 1, pp. 43–54, Jun. 2021, doi: 10.37339/e-komtek.v5i1.528.

[25] D. Yulistyanti, T. Y. Akhirina, T. Afrizal, A. Paramita, and N. Farkhatin, "Testing Learning Media for English Learning Applications Using BlackBox Testing Based on Equivalence Partitions," *Scope : Journal of English Language Teaching*, vol. 6, no. 2, p. 73, Apr. 2022, doi: 10.30998/scope.v6i2.12845.

[26] D. Yulistyanti, T. Y. Akhirina, T. Afrizal, A. Paramita, and N. Farkhatin, "Testing Learning Media for English Learning Applications Using BlackBox Testing Based on Equivalence Partitions," *Scope : Journal of English Language Teaching*, vol. 6, no. 2, p. 73, Apr. 2022, doi: 10.30998/scope.v6i2.12845.

[27] G. W. Sasmito and M. A. Mutasodirin, "Black Box Testing with Equivalence Partitions Techniques in Transcrop Applications," in *2023 6th International Conference of Computer and Informatics Engineering (IC2IE)*, IEEE, Sep. 2023, pp. 53–58. doi: 10.1109/IC2IE60547.2023.10331562.

[28] Gilang Ryan Fernandes and Ika Mei Lina, "Boundary Value Analysis Testing Against Library Applications Using the Black Box Method as System Performance Optimization," *Jurnal E-Komtek (Elektro-Komputer-Teknik)*, vol. 5, no. 1, pp. 43–54, Jun. 2021, doi: 10.37339/e-komtek.v5i1.528.

[29] C.-H. Tsai, S.-C. Tsai, and S.-K. Huang, "REST API Fuzzing by Coverage Level Guided Blackbox Testing," in *2021 IEEE 21st International Conference on Software Quality, Reliability and Security (QRS)*, IEEE, Dec. 2021, pp. 291–300. doi: 10.1109/QRS54544.2021.00040.