

Web-based Accommodation Management System with Tenant Functionality via LINE Integration

Thanchat Janphakdisakul
Collage of Arts, Media and Technology
Chiang Mai University
Chiang Mai, Thailand
thanchat.jan@cmu.ac.th

Pathathai Na Lumpoon
Collage of Arts, Media and Technology
Chiang Mai University
Chiang Mai, Thailand
pathathai.n@cmu.ac.th

Abstract—This study presents a web-based accommodation management system integrated with LINE to improve accessibility for tenants and reduce administrative burdens for dormitory managers. Phase 1 development includes room management, tenant data, rental contracts, income-expense records, and LINE-based registration approvals. A comparative analysis of five existing systems guided the design for small-scale dormitories. User feedback was gathered through interviews, User Acceptance Testing (UAT), and post-interviews, ensuring the system meets real user needs and expectations. A case study of Mænuansri Dormitory in Thailand showed the system's ability to reduce errors from manual processes and improve information access. The system follows ISO 29110 standards, supporting maintainability and operational efficiency.

Keywords—accommodation management, dormitory system, LINE integration, tenant management, web application, ISO 29110

I. INTRODUCTION

The management of rental accommodations, particularly dormitories, is essential for providing long-term housing solutions to individuals who require stable and affordable lodging. Efficient dormitory management plays a crucial role in ensuring tenant satisfaction and maintaining operational order. Traditionally, dormitory management has relied on paper-based records or standalone spreadsheet applications, which pose significant limitations in data storage, retrieval, and communication. These methods often result in human errors in rent calculations, misplaced documents, and disputes over rental agreements and maintenance requests [1].

User research conducted through interviews with dormitory managers and tenants revealed common issues, including incorrect rent notifications, miscalculations of rental fees, and damage deposits disputes due to tenants' lack of access to rental contract details. Tenants expressed a preference for a system that integrates with existing communication applications to facilitate seamless interactions while ensuring data security. Despite the availability of various accommodation management systems, many do not fully address the needs of small-scale dormitory owners, particularly the requirement for integration with widely used chat applications and the flexibility to customize features based on specific business needs. This research aims to develop an efficient and scalable dormitory management system that enhances communication, reduces administrative workload, and improves tenant experience.

To address these challenges, this research focuses on developing a web-based accommodation management system integrated with the LINE platform, which is widely used in Thailand. LINE is already a primary communication tool between dormitory managers and tenants, making it an ideal choice for seamless interaction and notification delivery.

To develop this system, a comparative study of five existing accommodation management systems, as shown in Fig. 1, was conducted to identify best practices and areas for improvement.

System	Horganice	Apartrent	Yeeraf	Apartmentery	Horpak	This Project
Dormitory Administrator Side	Web, Mobile	Web	Web	Web	Only for Android	Web
Room Management	✓	✓	✓	✓	✓	✓
Meter Recording	✓	-	✓	✓	✓	✓
Bill Generation	✓	✓	✓	✓	✓	✓
Lease Contract Creation	✓	✓	✓	✓	✓	✓
Parcel Management	✓	✓	-	-	-	-
Dashboard Summary	✓	✓	-	✓	-	✓
Reports	✓	✓	-	✓	✓	✓
News Notification	✓	✓	-	-	-	✓
Tenant Side	Mobile	Web	-	-	Only Android	LINE
Bill and Lease Contract	✓	✓	-	-	✓	✓
Payment Notification	✓	✓	-	-	✓	✓
News Notification	✓	✓	-	-	-	✓
Problem Reporting	✓	✓	-	-	-	✓
Room Fee Analysis	✓	-	-	-	-	-
Credit Card Payment	-	✓	-	-	-	-
Support and Consultation Team	✓	-	-	-	-	✓
Differences	There's an extra service connected to accounting system.	There's a credit card payment system.	There's lease termination issuance system.	There's important event notifications via email and LINE.	There's repair and maintenance records.	Utilize LINE for dormitory-related operations.
Free Trial	30 days	Out of service since 2020	90 days	90 days	60 days	-
Software Pricing	Start 679 baht per month		Start 9 baht per room per month	Start 660 baht per month	-	-

Fig. 1. Comparison of dormitory management system

A comparative study of five dormitory systems revealed that most supported desktop platforms, with only one mobile-only and one cross-platform system. All included room, billing, and contract management, while four offered utility tracking, dashboards, and rental reports—highlighting six key features essential for effective dormitory management.

Despite their strengths, such as tenant handling, billing accuracy, maintenance tracking, and business reports, limitations were found in pricing models (subscription-only) and lack of customization. These gaps emphasize the need for a flexible, one-time-purchase system tailored to diverse user needs—especially for small-scale dormitories still relying on manual processes.

The system follows the Software Development Life Cycle in accordance with ISO 29110, ensuring a structured and standardized approach. It is divided into three phases, with this work focusing on Phase 1, which covers fundamental dormitory data management, room administration, tenant information, rental contracts, and financial record-keeping, as illustrated in Fig. 2. The proposed system is expected to minimize human errors in rental transactions, improve document accessibility, and facilitate seamless communication through LINE integration. Additionally, it enhances operational efficiency for dormitory managers by automating key processes such as rental notifications, contract storage, and tenant management.

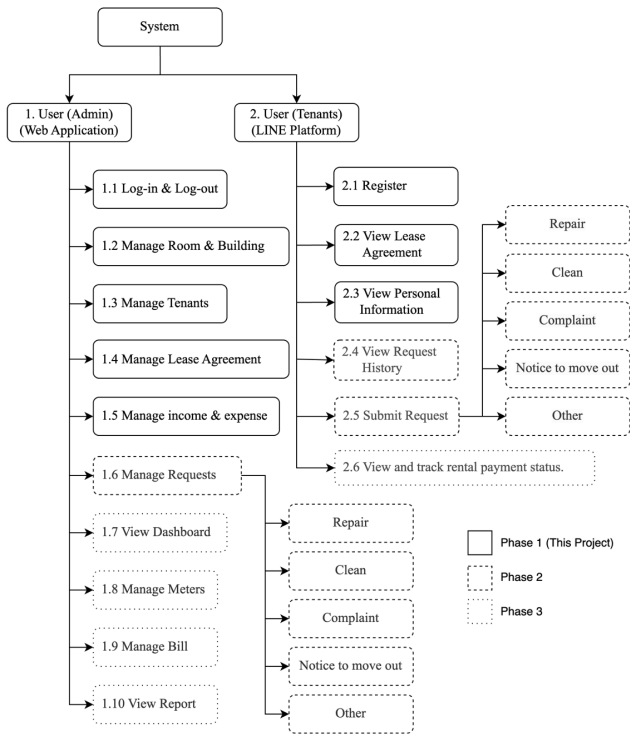


Fig. 2. Overview of System Functions and Development Phases

To ensure long-term maintenance and scalability, project documentation is maintained for traceability, quality assurance, and system validation. It includes the Project Plan, Software Requirements Specification, Product Design Document, Configuration Management Plan, Test Plan, Test Record, Traceability Record, and User Manual with a Video Guide.

This study proposes a web-based dormitory management system integrated with LINE for small-scale operations. Developed using user-centered design and ISO 29110 standards, the system ensures practicality and maintainability. The paper is structured as follows: Section II reviews related work; Section III details the system architecture; Section IV covers implementation; Section V presents a case study; Section VI discusses findings; and Section VII outlines future directions.

II. RELATED WORKS

A. Concepts and Theories

A dormitory is a residential facility for students and the general public [2]. According to Thai law, a dormitory provides rental accommodation under the supervision of a manager [3]. Dormitory management systems streamline administrative tasks, enhance communication, and improve efficiency by reducing reliance on manual records [4]. Effective systems should include tenant management, room tracking, billing, maintenance requests, online access to rental information, record keeping, and data security [5].

LINE is a widely used communication platform offering LINE Official Account (LINE OA), which enables businesses to interact with users through messaging, automated responses, and interactive menus [6]. Integrating LINE OA into dormitory management systems enhances communication, facilitating automated billing and notifications.

ISO 29110 is a software engineering standard ensuring structured development processes for small enterprises. It includes project management, software deployment, and documentation guidelines to enhance software quality [7].

B. Relevant Studies

SharmikhaSree et al. [8] developed a dormitory management system for university students in India, addressing inefficiencies in paper-based processes. The system utilized XAMPP, PHP, and MySQL, providing secure access and improving data organization.

Chye and Mubin [9] designed the APResidence online dormitory management system for Asia Pacific University in Malaysia. Built with Angular and PHP, it emphasized user authentication and streamlined communication between tenants and staff.

These studies highlight the importance of database-driven dormitory management and communication features. This research integrates LINE OA and ISO 29110 to enhance system efficiency and ensure structured software development.

III. SYSTEM DESIGN AND ARCHITECTURE

The proposed accommodation management system follows a web-based architecture, integrating a structured database and an external communication platform (LINE) for tenant interactions.

A. System Design

The system supports two primary user roles: dormitory managers (Admin Users) and tenants (Tenant Users). It was developed based on requirements from interviews and a comparative analysis of five existing systems. Core functionalities were defined by user roles and implemented using a user-centered design approach. UI/UX prototypes were created using Figma, as illustrated in Fig. 3. Use Case Diagrams and a Requirements Specification Document were employed to ensure traceability and alignment with user needs.

1) Admin User Features

a) *Login and Logout*: Admin users can securely authenticate using a username and password, and manage their login session, including logging out at any time.

b) *Room and Building Management*: Admin users can monitor the availability status of rooms, as illustrated in Fig. 4, and manage building data, including room types (fan-cooled or air-conditioned).

c) *Tenant Management*: Admin users can store tenant details, track rental history, and manage tenant status updates, as illustrated in Fig. 5.

d) *Contract Management*: Admin users can handle rental agreements, renewals, and the ability to download or print contracts in different formats (Word and PDF).

e) *Income and Expense Management*: Admin users can record, track, and manage financial transactions related to income and expenses, with options to search and delete specific records.

2) Tenant User Features

a) *Tenant Registration*: New tenants can register through the system by providing necessary personal details via a LINE-based registration form.

b) *Contract Viewing*: Tenants can access and view their rental agreements directly through the system interface, as illustrated in Fig. 6.

c) *Personal Information Management*: Tenants can view their personal details post-registration, with changes requiring dormitory manager approval, as shown in Fig. 6.

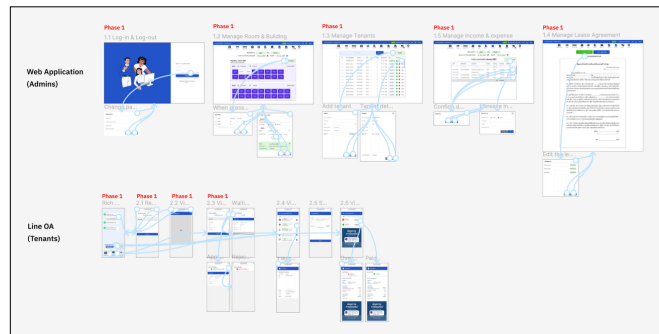


Fig. 3. UI/UX Design

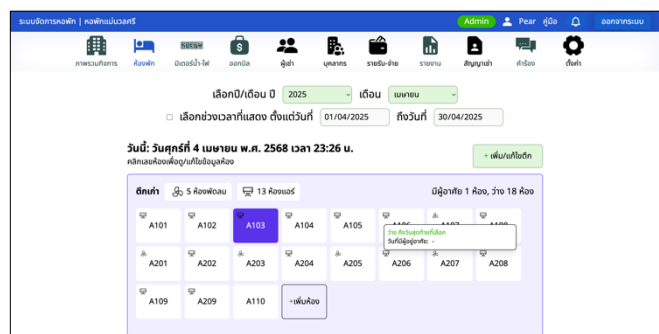


Fig. 4. Room page displaying room availability status when hovering over the room button.

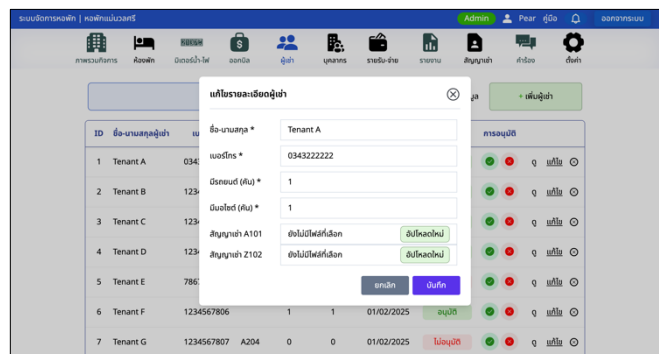


Fig. 5. Tenant page displaying a modal for editing tenant details.

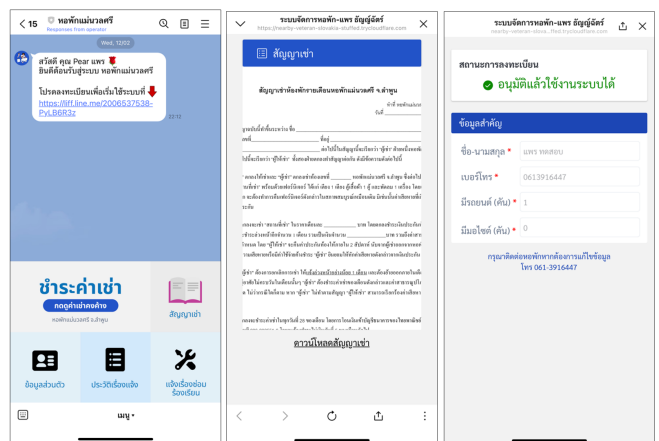


Fig. 6. Tenant-side interface displayed on mobile via the LINE application.

The architecture comprises the following key components:

- **Frontend**: Developed using Vue.js 3.0, the frontend provides an interactive user interface for both administrators and tenants.
- **Backend**: Built with Node.js, the backend handles business logic, data processing, and API interactions. It connects to the database via RESTful API, ensuring secure and reliable data exchange.
- **Database**: MySQL serves as the primary database, hosted within a Docker container for simplified deployment and management. phpMyAdmin is utilized for database administration.
- **LINE Integration**: The system leverages the LINE Messaging API to notify tenants about system approvals and other important updates, enhancing communication efficiency between managers and tenants. Additionally, the integration includes LINE Login and LINE LIFF to facilitate seamless authentication and in-app interactions. The system also utilizes LINE Official Account (LINE OA) and Rich Menu to improve accessibility and user experience.

The interaction between these components is depicted in Fig. 7, demonstrating how data flows seamlessly from the frontend through the backend to the database while incorporating external communication via LINE.

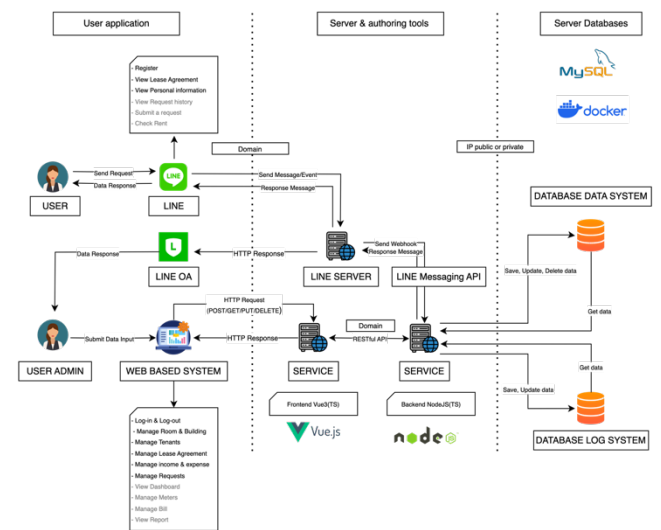


Fig. 7. System Architecture

B. Database Design

The database design follows core business rules to ensure consistency and traceability. Key entities include Admins, Buildings, Rooms, Rooms_Tenants, Tenants, Contracts, and Income_Expenses. The Admins table stores administrator credentials and permission levels. The Buildings table contains building names and floor information, while the Rooms table defines room numbers, types, and statuses. The Rooms_Tenants table functions as a booking log, capturing room occupancy history and current tenant assignments. The Tenants table stores personal and contact information for each renter. Each booking is linked to a single entry in the Contracts table, which contains rental agreement terms and durations. Financial transactions are recorded in

the Income_Expenses table and are associated with both Rooms and the responsible Admins.

Relationships are enforced through foreign keys: Buildings to Rooms (1:N), Rooms to Rooms_Tenants (1:N), Rooms_Tenants to Tenants (N:1), Room_Tenants to Contracts (1:1), and Rooms and Admins to Income_Expense (1:N and N:1 respectively). A data dictionary defines field types, constraints, and validation rules. Fig. 8 illustrates the Class Diagram, while Sequence Diagrams represent key interactions across the system.

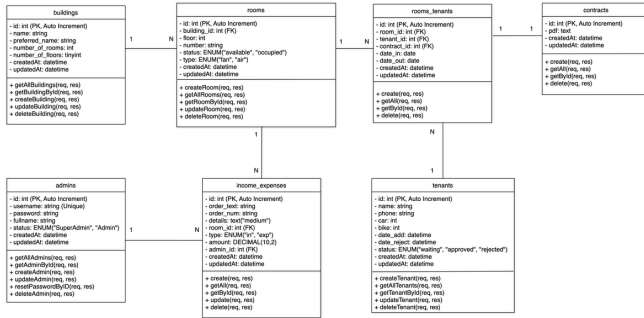


Fig. 8. Class Diagram

IV. IMPLEMENTATION

A. Development Process

The system was developed using Visual Studio Code as the primary development environment, with GitHub Desktop for version control. The project followed an Iterative Model, ensuring continuous refinement through multiple development cycles. The system was structured into two main components: Frontend (FE) and Backend (BE).

The development process started with the frontend implementation, followed by backend development and API integration. The backend was built using Node.js, with Sequelize as an Object-Relational Mapping (ORM) tool for managing the MySQL database. RESTful APIs were implemented to enable seamless communication between the frontend and backend.

API functionalities were tested using Postman to ensure data consistency and proper functionality. Each module underwent iterative testing and debugging, allowing for continuous system improvements, error minimization, and performance optimization before the final system evaluation.

B. System Testing and Evaluation

The system underwent unit testing, system testing, and user acceptance testing (UAT) to validate functionality, performance, and reliability. A total of 121 test cases were planned across all testing levels: 56 for unit testing, 29 for system testing, and 36 for UAT. Unit testing was conducted on individual components to verify expected results and outputs. System testing ensured seamless integration between the frontend, backend, and database, with outcomes documented based on actual results and pass/fail criteria.

UAT was performed using defined test scenarios and expected results, following a structured process that included test preparation, execution, result recording, system refinement, and retesting. Three participants, including dormitory administrators and owners, were involved in the evaluation. Additionally, post-test interviews with 12 questions were conducted to gather feedback for further improvements.

C. Documentation

The software development process adhered to ISO/IEC 29110 to ensure comprehensive and structured documentation. Key documents included the Project Plan, Software Requirement Specification (SRS), Software Design Document (SDD), Configuration Management Plan, Test Plan, Test Record, Traceability Document, and User Manual. The user manual was provided in both document format and video tutorials uploaded to YouTube. This documentation approach facilitated system maintenance, improved usability, and ensured effective project management.

V. CASE STUDY: MÆNUANSRI DOMITORY

This case study focuses on Mænuansri Dormitory, a rental property consisting of 34 rooms across two buildings. The study investigates existing dormitory management issues, the data collection process from administrators and tenants, the system's real-world implementation, and performance improvements after deployment.

A. Challenges in Dormitory Management

The traditional dormitory management approach relied on Microsoft Excel for rental calculations and manual processes for invoicing and communication via LINE. Several key challenges were identified:

- Frequent errors in rent notifications due to manual entry and incorrect recipient selection.
- Excel formula miscalculations, leading to incorrect billing amounts.
- Loss or damage of paper invoices, causing delays in rent payments.
- Lack of structured financial tracking, making it difficult to monitor revenue and expenses.
- Challenges in enforcing tenant agreements, as lease terms were often overlooked or disputed.

B. Data Collection from Administrators and Tenants

To address these challenges, exploratory research was conducted by interviewing the dormitory administrator and tenants.

- Dormitory Administrators highlighted operational inefficiencies in rent collection, billing, and contract enforcement.
- Tenants provided feedback on preferred system features, including LINE integration, repair request submission, and direct contact with administrators.

C. System Deployment and Results

To evaluate the system's effectiveness and usability, UAT and post-test interviews were conducted with real users from Mænuansri Dormitory. The testing involved three participants: two dormitory managers and one owner, aged 24, 63, and 58 years. The outcome of this testing focused on assessing user proficiency with the system, collecting feedback on the use of each feature, and observing real user behaviors. The testing also provided insights into the time required to complete each test scenario. Additionally, this process helped identify issues and errors that were not previously detected during earlier stages of development and testing, which were corrected and re-tested.

Since the system was still under development and not fully deployed, Cloudflare was used for deployment instead of relying solely on a local host. This decision was necessary to support HTTPS connectivity, which is essential for LINE service integration. As a result, the deployment was made more accessible and secure for the users while maintaining necessary functionalities for the LINE integration, even before the full system was completed.

D. Performance Evaluation Before and After

Before the development of the Mænuansri Dormitory Management System, dormitory management relied on manual processes using Microsoft Excel and paper-based documentation. This approach led to inefficiencies, including difficulty in tracking tenant records, scattered contract storage, and reliance on physical documents, which were prone to loss and human error. Additionally, the workflow required manual coordination, making it time-consuming and inconsistent.

In Phase 1, the system was developed to improve data management and accessibility, focusing on digitizing tenant records and contract storage. However, since the system is still under development, it has not yet been officially deployed for real-world use. Instead, UAT was conducted with dormitory managers and the owner to evaluate the system's usability and effectiveness.

The UAT confirmed that the system meets functional requirements. However, users, especially those less familiar with digital platforms, required time to adapt to the interface. Misclicks, such as selecting 'Save' instead of 'Add Tenant,' were common, indicating the need for clearer interface cues. Older participants took longer to complete tasks, though performance improved with repetition. Operational bottlenecks included slow data retrieval times due to Cloudflare Tunnel latency and previously undetected bugs. Navigation improvements and UI adjustments were suggested to enhance usability. Key features like billing and payment tracking were not included in Phase 1 and are planned for future development.

While system usability and stability were positively received, some users expressed a need for more structured tutorial materials, such as shorter instructional videos, to facilitate learning. The feedback collected from this testing phase will guide further refinements and enhancements in the upcoming phases before the system is fully deployed.

VI. RESULT AND DISCUSSION

During Unit Testing and System Testing, 13 test cases initially failed, but all identified issues were resolved and verified. In UAT, three additional issues were found, particularly affecting older users who required more time to navigate the system. After implementing necessary fixes, a follow-up test was conducted with users, and all cases successfully passed. Feedback from the UAT also provided valuable insights into user experience, system usability, and areas for improvement.

The testing process highlighted the system's effectiveness in managing tenant data and streamlining operations, but it also revealed challenges related to user adaptability and interface clarity. Some users, particularly those unfamiliar with digital systems, needed additional time and guidance to fully utilize the platform. While the system improved

accessibility and data management, further refinements in UI design and user experience are necessary to enhance usability. Additionally, users are still waiting for the completion of Phase 2 and Phase 3 so they can fully utilize the system in real-world operations soon.

VII. CONCLUSION AND FUTURE WORK

The development and testing of the web-based accommodation management system with LINE integration demonstrated its potential to improve tenant management efficiency. The system successfully digitized key management tasks, enhanced accessibility for dormitory staff, and streamlined communication with tenants. Despite these improvements, the testing phase revealed usability challenges, particularly for older users, which were addressed through iterative refinements. The system is not yet fully deployed, as further development is required to complete all planned features across three phases.

Future development will focus on completing Phases 2 and 3, which include key functionalities for both the administrator and tenant sides. For the administrator, Phase 2 will introduce the feature for managing tenant requests, while Phase 3 will include functionalities for business overviews, water and electricity meter management, bill generation, and report generation. On the tenant side, Phase 2 will include features for viewing request histories and submitting new requests, while Phase 3 will enable tenants to track and monitor their rent payment status. Enhancements in user interface design will also be prioritized to improve accessibility, especially for users unfamiliar with digital platforms. Additionally, real-world deployment and long-term performance evaluations will be conducted to assess system stability, scalability, and overall impact on dormitory operations.

REFERENCES

- [1] B. A. Weber, H. Yarandi, M. A. Rowe, and J. P. Weber, "A comparison study: paper-based versus web-based data collection and management," *Applied Nursing Research*, vol. 18, no. 3, pp. 182–185, Aug. 2005, doi: 10.1016/j.apnr.2005.01.001.
- [2] Ratchabandit Society, *Pojjanukun Jappab Rajabundit Yasathan*, 2011.
- [3] Office of the Council of State, *Act on Dormitories*, Available: <https://www.mhesi.go.th/images/2563/pusit/legal-all/66p2558.pdf>, 2015.
- [4] Smart Living, "How a good dormitory management system works and its key features," Available: <https://www.kobkid.com/เรื่องน่ารู้/ระบบจัดการหอพักคืออะไรและมีฟีเจอร์อะไรบ้างที่น่าสนใจมาดูกัน>, Mar. 15, 2022.
- [5] SpaceBasic, "12 essential features of a hostel management system," Available: <https://www.linkedin.com/pulse/12-essential-features-hostel-management-system-spacebasic-inc--rlpdc/>, Jul. 26, 2023.
- [6] LINE, "Comparison between official and personal accounts," Available: <https://lineforbusiness.com/th/service/line-oa-features>, accessed Sep. 12, 2024.
- [7] International Organization for Standardization, *Software engineering — Lifecycle profiles for Very Small Entities (VSEs) — Part 5-1-2: Management and engineering guide: Generic profile group: Basic profile (ISO/IEC TR 29110-5-1-2:2011)*, Geneva, Switzerland: ISO, 2011.
- [8] SharmikhaSree, R., Meera, S., Kavya, K. K., & Harina, P., "Dormitory management system," *Intelligent Computing and Control for Engineering and Business Systems (ICCEBS)*, IEEE, 2023. Available: <https://ieeexplore.ieee.org/document/10449199>, 2023.
- [9] T. C. Chua and S. A. Mubin, "APResidence: Development of online student accommodation management system for Asia Pacific University," *3rd International Conference on Mobile Networks and Wireless Communications (ICMNWC)*, IEEE, 2023. Available: <https://doi.org/10.1109/ICMNWC60182.2023.10435913>, 2023.