JIT Mahasarakham University, Thailand
http://jit.it.msu.ac.th

Journal of Applied Informatics and Technology, 7(2), 252-267.

Research Article

An In-House Time Tracking Application Development with a Low-To-No-Code Platform

Sorn Tanharaphan[1], Thanabadee Rodvanich[1], Yatawee Aupasao[1], Natsuda Kaothanthong[1,*]

[1] School of Management Technology, Sirindhorn International Institute of Technology, Thammasat University, Pathum Thani 12000, Thailand

* Corresponding author: Natsuda Kaothathong, natsuda@siit.tu.ac.th

Received:
20 November 2023
Revised:
22 January 2024
Accepted:
2 April 2024

Keywords:

AppSheet, Human Resources Management, Human Resources Management Software, Low-Code/No-Code Platforms, SMEs

Abstract: We demonstrate a procedure exhaustively to develop an in-house application without requiring database establishment and technician using a low-to-no-code (LCNC) platform that aligns with the organization's business case for tracking working time and leave request of employees. We developed the application using Google AppSheet and Google Sheets for data storage. The proposed application has three main functions to accurately collect a daily clock-in, clock-out, retrieve the total number of working hours in the current month, and deal with leave requests that align with the organization's regulations. Furthermore, there are three functions for an administrative employee manage the staff's information, generate a monthly work report, and perform leave request approvement. The application has been tested at the Artificial Intelligence Association Thailand (AIAT), where time tracking and leave request were previously managed with paper-based. The usability test shows that the staff takes on average less than 1 minute to perform a daily clock-in and clock-out, less than 2 minutes to complete a leave request as well as cancel the submitted request. The learnability test shows that the users take less than 1 minute to complete the daily working timestamp and send leave requests.

1. Introduction

Human Resource Management (HRM) is an essential part of businesses, regardless of their size. HRM focuses on activities like recruiting, hiring, deploying and managing an organization's employees. To help simplify the HRM processes, companies can utilize the Human Resource Information System (HRIS) to assist in the storing and processing of employees' data without needing to spend too many resources. Available software in the market costs ranging from $20-$40 per user per month for a basic plan to $80-$100 per user per month for a pro plan (Empeo, 2024; Gusto, 2024). Plus, the adoption of new technologies comes with switching costs, which include not just the software but also associated expenditures like learning and training (Avlonitis & Panagopoulos, 2005). Due to these costs of information system acquisition, some organizations still apply a traditional method with paper-based, which is resource-intensive. In this work, we focus on time recording and leaves submission functions in HRM.

Many low-to-no code platforms have been introduced to facilitate non-technical users. Data analysis software such as Power BI, Tableau, and RapidMiner were introduced to allow an end-user to complete the task by drag and drop procedures without requiring any complicated coding. With user interfaces and understanding of data, the users are able to produce a dashboard for a descriptive analysis and perform a simple predictive analysis. In addition, software for simple data management from data entry, data storage, and data retrieval was proposed. Application development software like Power Apps and Google AppSheet were proposed for building cross-platform applications. With a connection to the available database or spreadsheet, the users can create their own application that can be executed on mobile devices and web sites. One advantage of low-to-no code platforms is that the end users are able to create their own applications and the cost of software or applications acquisition is lower than a software development or installation.

This work demonstrates how a low-to-no code application is applied to implement simple function of HRM without establishing a company database. The developed application can facilitate employees recording working hours, and submission of absence requests, which is one of the functions in HRM. The proposed application is built using the Google AppSheet which is a no-code platform. The development of the application requires low technical knowledge in programming and can be integrated well with widely used tools such as spreadsheets. The completed application has been implemented and tested with a small business. The result shows that the learnability and the usability of the proposed is satisfactory.

2. Previous Work

Human Resource Management (HRM) is the strategic strategy which firms employ to manage their employees efficiently. HRM

includes the policies, processes, and systems that are set up to recruit, instruct, and manage company personnel. To perform HRM, a human resource information system (HRIS) performs an essential part in facilitating the effective integration of technology solutions (Thite, Kavanagh, & Johnson, 2012). The main purpose of HRIS is to ensure data integrity, provide up-to-date information, and keep track of the employees that the company can use to support daily operations such as monitoring personnel work time and leave requests. Applications of HRM have been incorporated through HRIS to assist HRM at different levels within an organization. While HRIS primarily focuses on the technological aspects of managing HR information, a human resource management system (HRMS) takes a step further by integrating these technologies into a more comprehensive framework. HRMS not only ensures data integrity and real-time information but also actively contributes to the strategic management of human resources.

Low-code and no-code (LCNC) data analysis platforms offer advanced analytical capabilities, seamless data integration, and interactive visualization tools. These platforms can help users make the data analysis processes more efficient since manual efforts are less required to complete each task, making it easy for data-driven businesses seeking for insights without relying only on IT expertise. For example, Power BI, as a LCNC platform, is available for users to perform data analysis without extensive knowledge in programming. However, the limitations of the LCNC platforms include vendor lock-in concerns, complex use cases handling, and a lack of on-premises support (Sufi, 2023).

Low-to-no-code platform expedites the development of applications by reducing the amount of coding and resources associated with building an application from scratch. For businesses, this technology helps in terms of speed to market and little cost of creating software (Sattar, 2018). One of the low-code platforms is Microsoft Power Apps. Power Apps combines the positives of low-code development with the capacity of integration with Microsoft products such as Excel. Despite that, one of its setbacks comes in the form of subscription cost; since to use Power Apps, the user needs a Microsoft Dynamic subscription with the add-on of Power Apps billing. On the contrary, Google AppSheet offers similar services at a lower subscription fee. Platform called OutSystems is a user-friendly interface that can be developed using pre-built templates. The platform enables across devices accessibility which helps developers with streamline delivery of mobile and application (Martins *et al*., 2020). The developed applications using a low-code platform has advantages of minimal coding and short time to market. However, the vendor-lock effects lead to challenges of governance plan and design limitations. Though the low-code development platform possesses limitations, it is applicable for the secondary business activities (Ledl, 2022).

3. Material and Method

In this work, we adopt the business rules of one small association in Thailand to develop the application accordingly. We employ cloud software called AppSheet to develop a mobile application. There are 3 main functions that allow a small business owner to perform an employee management: 1) working hours record, 2) leave request, and 3) a monthly report. The application applies spreadsheets to read and record the data.

To implement the HRM management software with AppSheet, spreadsheet files for recording employees' profiles, working hour reports, leave availability and transactional data are required. The entity relationship diagram of the organization is depicted in Figure 1.

3.1 Data

The HRM of organization relates to 6 entities that are 'Staff', 'Time_record', 'Leave_request', 'Leave_available', 'Working_hour_report', and 'Generate_work_hour' as shown in Figure 1. The record of each entity is stored in a spreadsheet file. The 'Staff' file stores the record of each employee that are an employee ID, name-surname, phone number, and start working date. The employee ID is used as a key.

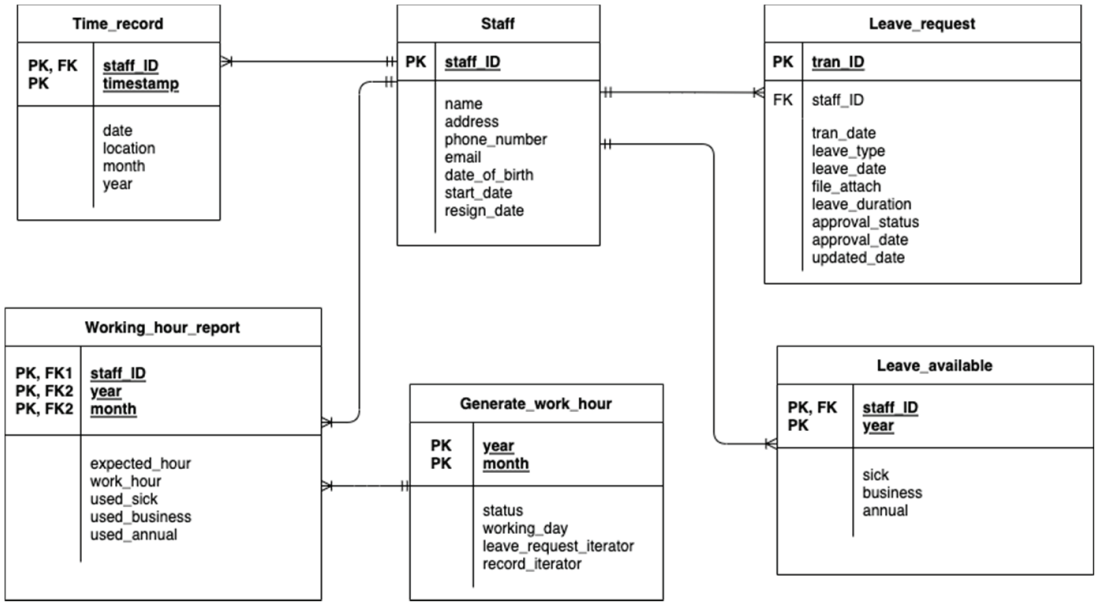


Figure 1. Entity relationship diagram

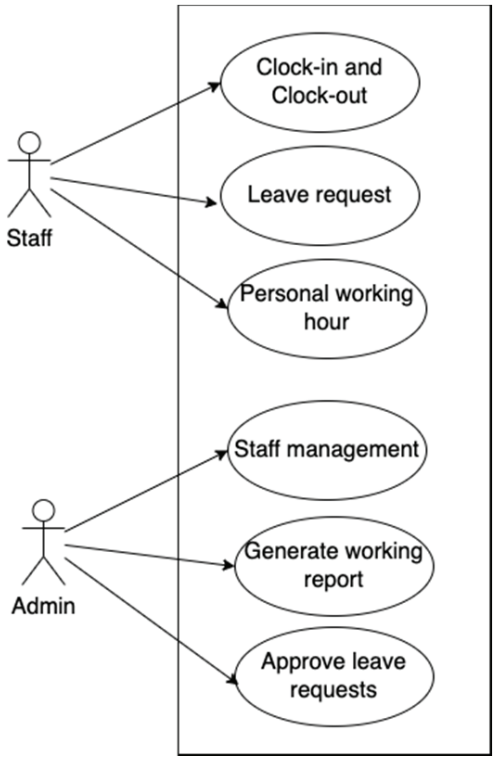


Figure 2. Application's use case diagram

For each working day; date, clock-in time, clock-out time, and location entered by each employee are recorded in the daily working hour file called 'Time_record'. This file is a transactional data that collects timestamp records when staff clock in and clock out. The organization's employees have 3 types of leave: 1) annual leave, 2) sick leave, and 3) business leave. The file 'Leave_request' collects transaction of requests submitted that record the data of transaction ID, employee ID, leave duration, leave type, and approved status. The number of leave days available for each employee can be retrieved from the file 'Leave_available'. The records in the spreadsheet files 'Time_record' and 'Leave_request' are used to calculate the total working hours of each employee at the end of the month and stored in the file 'Working_hour_report'. The file 'Generate_working_report' is a master file for generating the working hour report since it stores the required number of working days of each month.

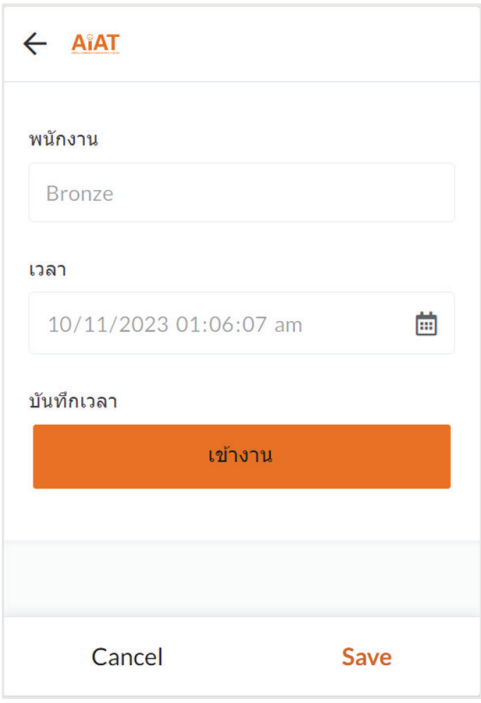
3.2 Application Development

3.2.1 Users

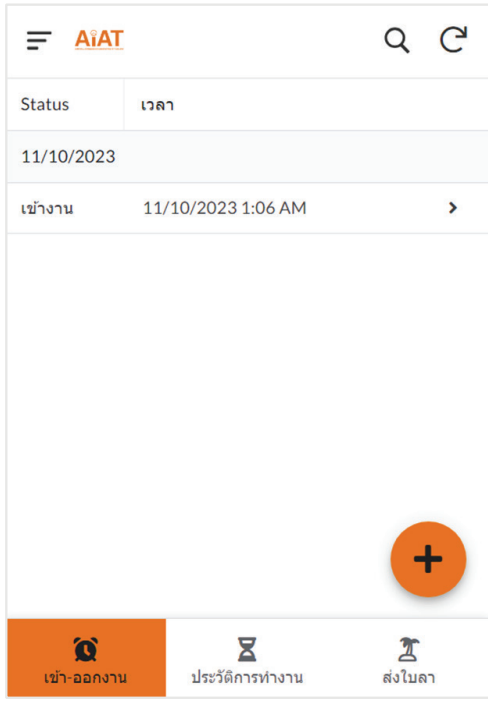
For data access control, the application manages data access by enabling users to both retrieve and update specific data according to their assigned roles. The application has 2 roles of users: staff and admin. Staff is a normal user as depicted in Figure 2. Admin is a super user, who can approve the leave request, generate a monthly report, and edit some data files that store the records. Staff is a user who can access the working time stamp, submitting leave requests, and retrieve his working hours.

3.2.2 Function

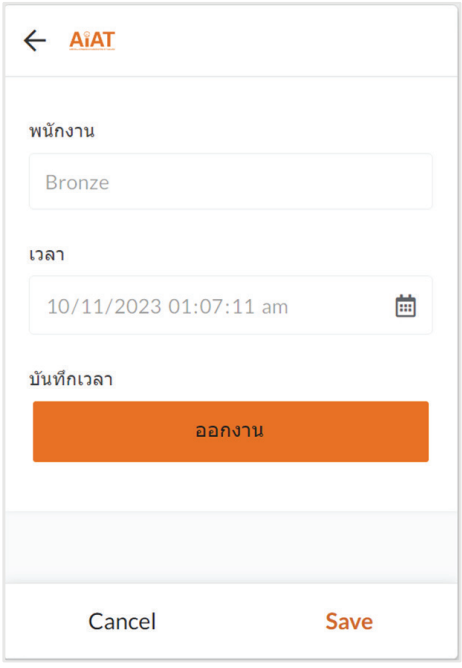
Our designed application has six functions in total as shown in Figure 2. Three functions are used by a normal user: 1) clock-in and clock-out, 2) leave request, and 3) personal working hours. The other three functions are used by a super user: 1) staff management, 2) generate working reports, and 3) approve leave requests. The detail of each function is described below.



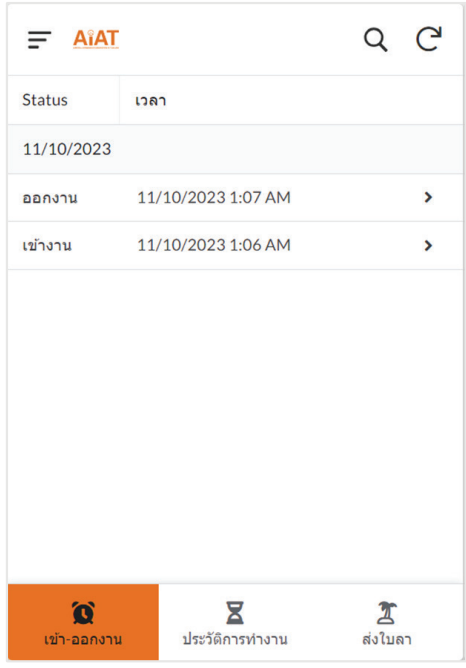
(a) Clock-in function



(b) Clock-in history



(c) Clock-out function



(d) Clock-out history

Figure 3. User interface of (a-b) clock-in function and (c-d) clock-out function of each staff

3.2.3 Clock-In and Clock-Out

The aim of this function is to ensure that the date and time entered at the time of the clock in, clock out, and location of each member of staff. The values of every employee are recorded in the same file named 'Time_record'.

To ensure the integrity of working hours such as the time entered the system, a staff press a button labeled as 'clock-in' when arrived the company and 'clock-out' before leaving then the timestamp and location are automatically inserted into the record. The user interface of the function is shown in Figures 3(a) and 3(c). The values of current date, clock-in time, clock-out time are automatically retrieved from the system and saved into the file with the employee ID. To ensure that the employee is in the office premises, the value of location is also recorded in the file 'Time_record' but it will not be seen by the staff. The clock-in and clock-out time of each day? can be checked by the staff on the application as depicted in Figures 3(b) and 3(d).

3.2.4 Leave Request

The leave request feature is available for users to submit leave requests to the system. Then the leave requests will be proceeded further by the admin. Figure 4 shows an interaction diagram for managing leave requests (Adamu, 2020). When the request was submitted by the staff, it can be canceled if it has not been approved by the admin. If the request has been approved, only an admin can cancel the approved request.

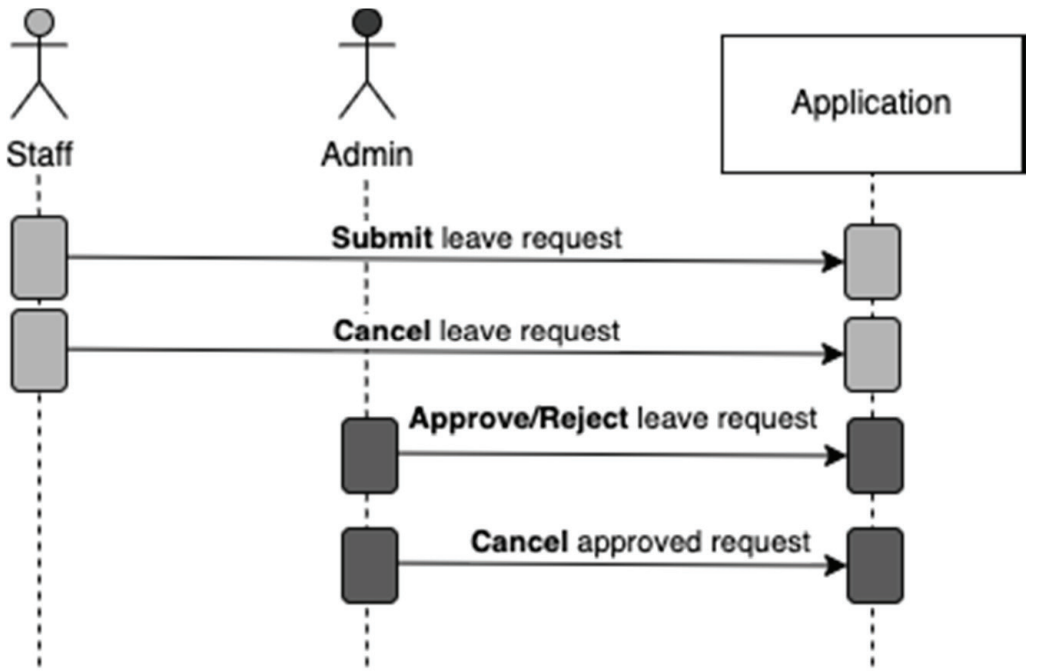


Figure 4. An interaction diagram for managing leave requests

To submit a leave request with the developed application (Fortich & Marcial, 2014), the users will begin with the "+" button and it will lead the users to the request form. This form requires users to provide specific details, including the type of leave (e.g., sick, annual, or business), the start date of the leave, and the leave duration. The users will be informed of their leave availability, ensuring it remains within the available leave balance. If the leave duration exceeds the leave balance, the warning message will be shown, and the user cannot submit the leave request until the error is fixed. The conditions for each leave type are as follows:

- For sick leave, the leave date should be either the current date, yesterday, or before yesterday. Requests must be submitted within two days after the sick leave date. If the sick leave is two days or longer, the users will be required to attach medical evidence as shown in Figure 5(a).

- For annual leave, the users must submit requests at least two days in advance without attaching any files as depicted in Figure 5(b).

- For business leave, the users are required to submit requests at least two days in advance and must attach a business leave document before submission in any case as represented in Figure 5(c).

When a leave request is submitted, it will initially display a "Pending" status. According to the approval status, there are four types of results that will be shown on the "Send Leave Request" page including "Rejected", "Canceled", and "Approved" status. While a request is in the "Pending" status, users have the option to cancel it before the admin decides. If a user cancels their request, the status changes to "Canceled."

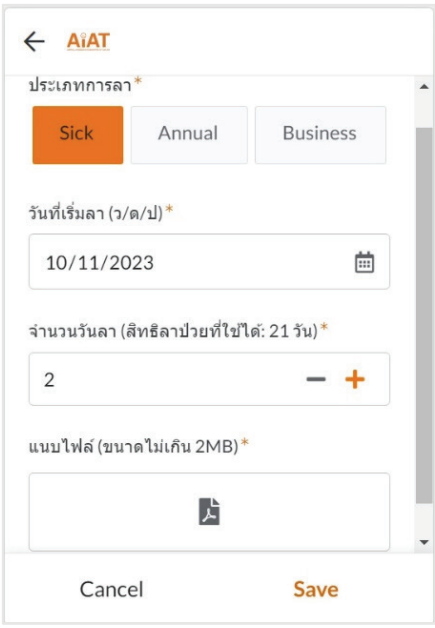
3.2.5 Personal Working Hours

This feature allows staff members to access their individual work record report in a particular month and year. After the admin generates the monthly work report at the end of each month, the new monthly report of each employee will show up on the personal work report page. In this page, the data is retrieved from the 'Working_hour_report' sheet, showing the expected and actual working hours of each month. Also, the users can review the number of leave days used in each month as shown in Figure 6.

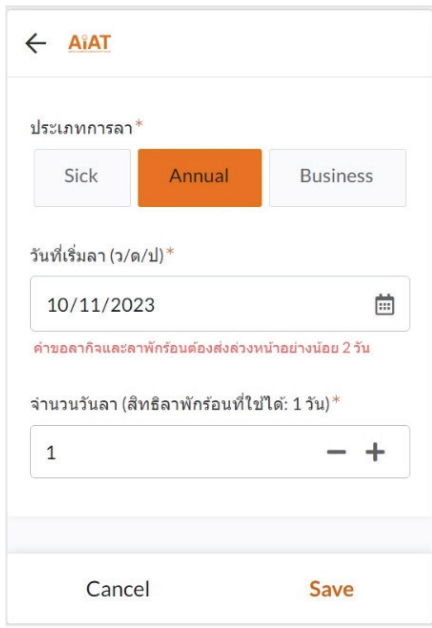
3.2.6 Staff Management

These functions are restricted to the admin and revolve around supervising staff's information. Staff records are stored in the 'Staff' file. Actions to manage information of staff include adding, viewing, and editing. Figure 6(a) shows the list of existing staff. On each row, there are shortcut buttons for editing, looking for staff's address using Google Maps, e-mailing, making a phone call, and sending a text message.

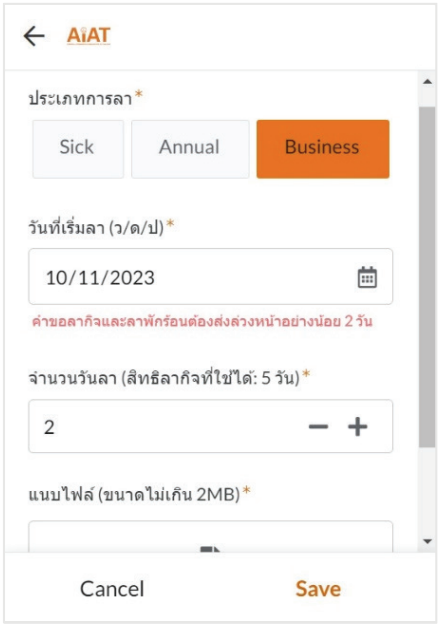
To add a new staff, tap the plus icon. It will bring up the form in Figure 7(b). Information required to create a new staff record comprises of ID, name, address, phone



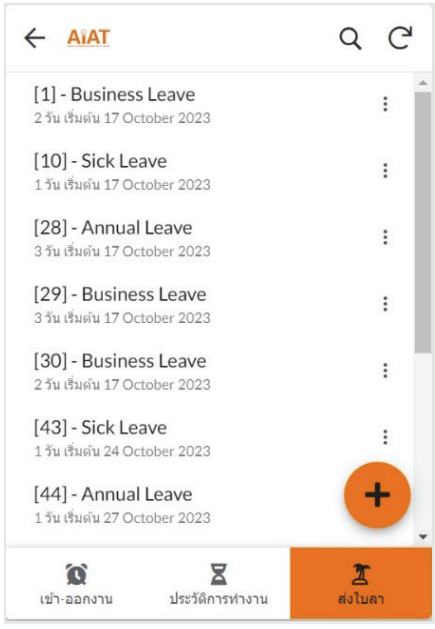
(a) Sick leave request



(b) Annual leave request



(c) Business leave request



(d) Leave request history page

Figure 5. User interface of submitting leave request of each type. (a) Sick leave, (b) Annual leave, (c) Business leave, and (d) request history

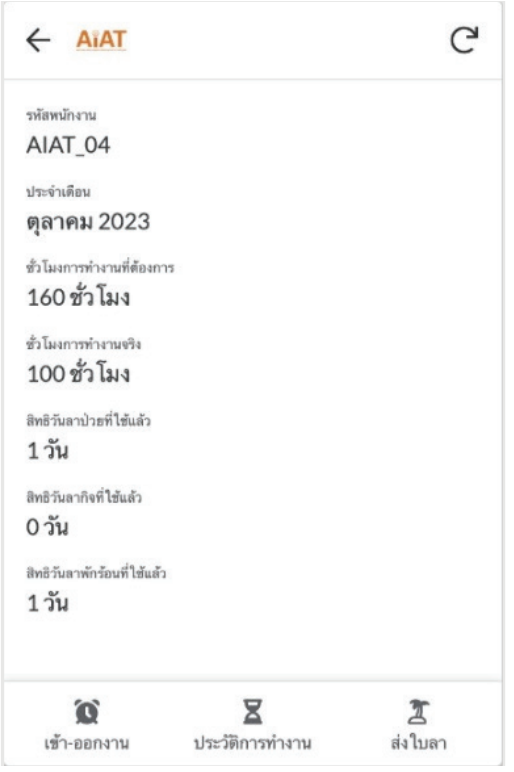


Figure 6. Personal work record

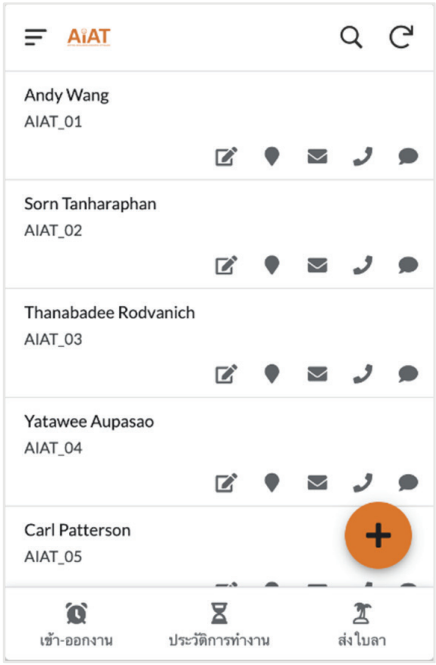
number, email, date of birth, and date of start work. The staff ID is automatically populated by incrementing the last staff's ID by one. For staff to be able to use this application, the inputted email must match the email the staff uses to log in. When a new staff is added to the 'Staff' file, an automated script will be called to generate a record in the 'Leave_available' data file telling how many days that staff has for sick, business, and annual leave. To view information about the staff, tapping the record of the staff in Figure 7(a) navigates you to the detail screen shown in Figure 7(c). For editing, the admin can either tap on the first shortcut icon on the staff record in Figure 7(a) or enter the Figure 7(c) detail screen and tap the floating edit button. Both will bring up the edit screen depicted in Figure 7(d).

3.2.7 Generate Working Report

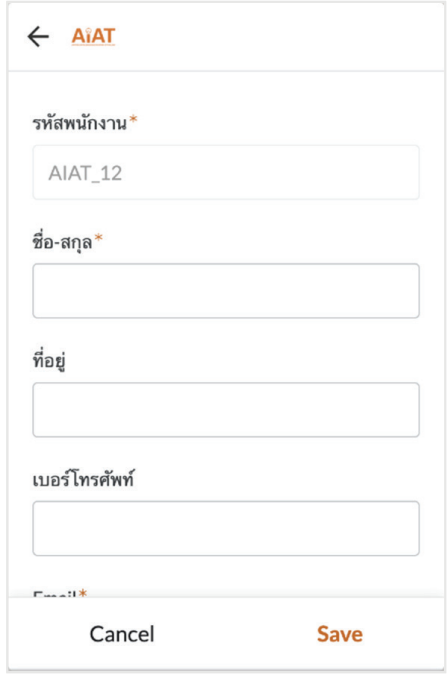
This function generates monthly working reports for all staff. The related data files are 'Generate_work_hour' and the 'Working_hour_report'. The purpose of 'Generate_work_hour' is to manage the creation of working reports of staff in specific months and years with the status of either Generated or To Generate. The 'Working_hour_report' is used to hold the generated reports.

This function executed JavaScript code in the Google Apps Script. To generate the staff's working report for the previous month, the admin clicks the add button on the bottom-right corner in Figure 8(a) which brings up the form shown in Figure 8(b). Inside the form, there are 3 input fields: month, year, and number of working days. Despite that, the admin is required to fill in only the number of working days. After the number-of-working-day field is entered, and the Save button is clicked, it triggers the JavaScrip written in the Google Apps Script. However, if there is not report generated in last two months which has the status of 'To Generate' in the record, the application does not allow the admin to proceed.

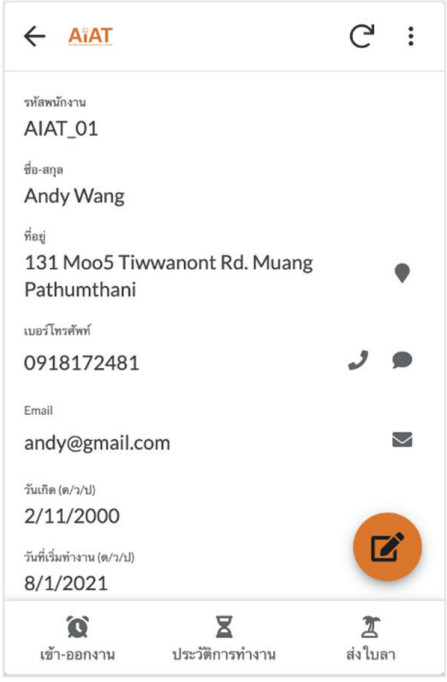
The Apps Script's function iterates over 'Leave_request', 'Staff', and 'Time_record'. The 'Staff' is used as the dimension table with staff ID as the primary key. Firstly, the function



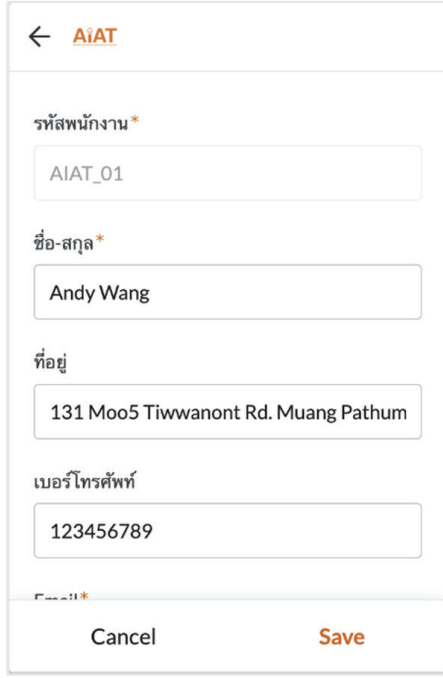
(a) Staff management page



(b) Add new staff page



(c) Staff detail page



(d) Staff edit page

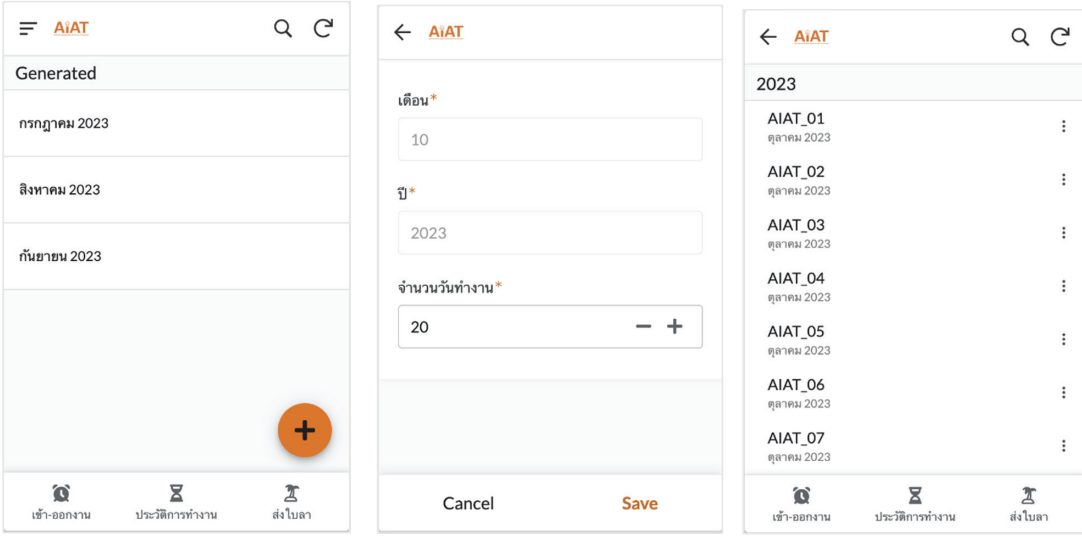
Figure 7. Example of user interface of functions that can be accessed by admin of the organization. (a) Viewing every staff detail, (b) Adding a new staff, (c) Detail of each staff, and (d) Editing information of each staff

goes over 'Leave_request' to sum up the approved leave date based on leave types by all staff. If there is a record of the month the admin is generating the report having the status of Pending, the process ends and notifies the admin through email to approve or reject that request. Otherwise, the process continues to iterate over the 'Time_record' to calculate the working time of staff. After finishing the process, the function now has the number of leave days of each leave type and the working time of all staff. It then inserts these records into 'Working_hour_report'. After new records generated from the Apps Script function were appended, the admin can view all staff's working hour reports shown in Figure 8(c).

3.2.8 Approve Leave Requests

This enables admins with the ability to accept, reject, and cancel leave requests made by normal users. The function will be available only for super users' applications.

As depicted in Figure 9(a), it will display the following features: Pending, Approved, Rejected, and Canceled. Admin clicks the 'Pending' option to view leave requests made by normal users following to Figure 9(b). Next, choose the user's request to modify the requested status by clicking on it. It will lead admin to the page as seen in Figure 9(c). By selecting the "edit" button in the bottom right, the admin can modify the desired status and also update the desired status. The status will indicate 'Pending' as a default value. Only super user may modify a status to 'Approved' or 'Rejected.' The status cannot be changed from "Pending" to "Canceled". The program will show a mistake as anything that appears in Figure 9(e). The status will change when the admin clicks the

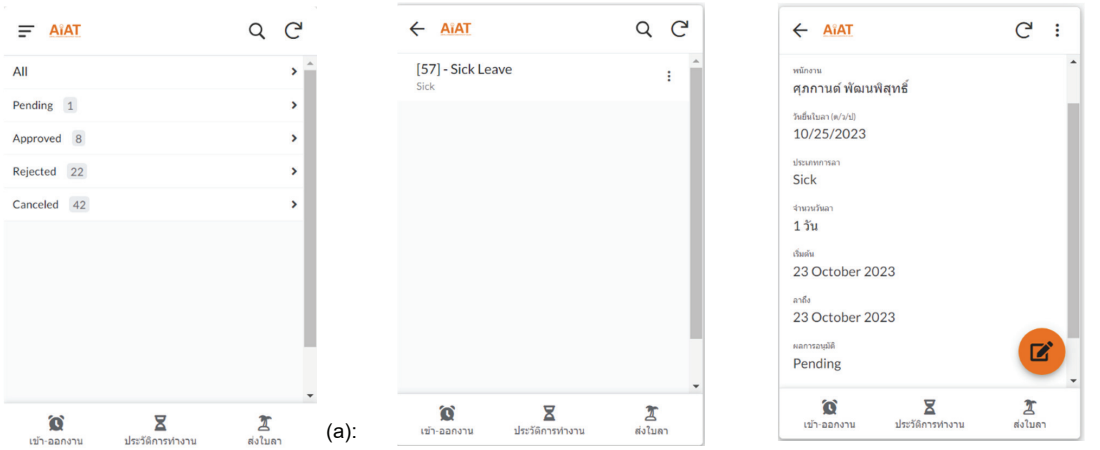


(a) Generate working report page

(b) Generate new working report page

(c) Working hour report page

Figure 8. User interface of monthly working report generation



(a): Approve leave request page (b): List of pending page (c): Information page

Figure 9: User interface of approve leave request method

'save' button. The date of the leave will be deducted from 'Leave_Available' sheet. For the 'canceled' status, it can be available only when normal user wants to cancel the leave request with 'approve' status only. To cancel a request, the administrator will select the 'Approve' option in Figure 9(a), click on the request, select "Edit", change the status to 'Canceled', and then click "Save". The leave date will be restored to the 'Leave_Available' sheet and the status will change to the 'Canceled'.

4. Results and Discussion

4.1 Results

The result of the application is assessed by its usability. After the development phase, the application went through testing by the business staff members. We evaluate three firm customers in varying age ranges (21–30, 31–40, and 41–50), in sequence, based on Table 1. Before the actual testing, staff members had around 15 minutes to experience and navigate functions inside the application. The main test cases are clocking in and out; sending and canceling leave requests; and viewing the details of work reports. The clock-in-clock-out function was fulfilled by all test pilots easily. The staff were also asked to carry out the task of sending three types of leave requests. These actions were done in under 2 minutes. The last test case is to view the individual working report where testers navigate effortlessly. However, the AppSheet limitation of no background refresh put a hindrance for test persons in using the application by leaving the status unchanged after the action was completed.

Table 1. Usability test and learnability test with 3 users in the company

Case No.	Task	Test Result	User 1 (Sec)	User 2 (Sec)	User 3 (Sec)	Average (sec)
	Usability Test					
1	Click Clock-in	PASS	5	6.2	5.2	5.8
2	Look at the clock-in history from Case No. 1	PASS	4.6	15	8.9	9.5
3	Click to see working report	PASS	2.8	6.7	5.5	5
4	Users ask for sick leave for 2 days, counting as the testing day is first day of leave, and attached file which is not more than 2mb	PASS	51.6	136.	82.2	90
5	Users ask for annual leave for 2 days in the next week	PASS	31.7	80	68.3	60
6	Users ask for business leave for 3 days in 2 days advance with attached file (file size must less than 2mb)	PASS	21	44.3	24.7	30
7	Users cancel their own leave requests which are still on pending status.	PASS	42	75.2	56.8	58
	Learnability Test					
8	Click Clock-out	PASS	3.2	5.4	4.1	4.2
9	Look at the clock-out history from Case No. 8	PASS	1.3	3	1.7	2
10	Users ask for sick leave 2 days, 2 days before the testing day	PASS	3.3	11.7	9	8

4.2 Discussion

The results of this paper indicate that, for small businesses: (1) the development of an HRM application for in-house use using the LCNC platform is simple and does not demand high technical skills; (2) the application built with this technology is cost-effective in both development and maintenance; and (3) the finished application imposes a very low learning curve for new users. The application is able to perform essential functions of clocking, calculating working hours, staff management, and submitting leave requests. With Google Sheet as the database, it simplifies the process for small businesses in terms of database maintenance. Although this study presents the feasibility of utilizing the LCNC platform for the implementation of HRM applications, there are some limitations to the platform. Some of them are the trade-off between

streamlining the development process and the level of customizability; and concerns about the scalability of using Google Sheet as the database.

For future research, it could explore the use of alternative LCNC platforms that offer similar benefits plus additional customizability and scalability. The results section presents the raw data, the experimental data, and the results after applying the techniques outlined in the methods section.

5. Conclusion

In summary, the HRIS implementation through Google AppSheet offers a cost-effective and efficient solution for small businesses. According to usability and learnability testing, its results confirm the user-friendliness and effectiveness of this application, proven by users performing the key tasks with ease, including clocking in and out, managing leave requests, and accessing working reports. The feedback from real users also emphasizes the application's efficiency in streamlining HR management. The integration of AppSheet with spreadsheets, along with its automated features, ensures data accuracy and reduces manual tasks. Despite its minor limitations, AppSheet, as a no-code platform, demonstrates its potential to enhance HR management and productivity. This work highlights the importance of low-code and no-code platforms in facilitating HR management to be accessible and practical for businesses of all sizes.

Acknowledgments

We would like to thank the Artificial Intelligence Association of Thailand (AIAT) for their supports and cooperations. This project is funded by School of Management Technology, Sirindhorn International Institute of Technology, Thammasat University.

Reference

Adamu, A. (2020). Employee Leave Management System. *Fudma Journal of Sciences*, 4(2), 86 - 91. https://doi.org/10.33003/fjs-2020-0402-162

Avlonitis, G. J. & Panagopoulos, N. (2005). Antecedents and consequences of CRM technology acceptance in the sales force. *Industrial Marketing Management, 34*(4), 355–368.

Fortich, M. S., & Marcial, D. E. (2014). Human resource e-leave management: A decision support system. *Informatika*: *Jurnal Teknologi Komputer dan Informatika*, *10*(2). https://doi.org/10.21460/inf.2014.102.329

Empeo (2024). *Pricing*. Retrieved 10 October 2023, Retrieved from https://www.empeo.com/en/pricing/

Gusto. (2024). Simple, transparent pricing. Retrieved 10 October 2023, Retrieved from https://gusto.com/product/pricing

Ledl, P. (2022). *Analyzing the potential of low-code platforms in digital transformation and implementing a project management application* (Master's thesis). Technische Hochschule Ingolstadt. https://opus4.kobv.de/opus4-haw/frontdoor/index/index/docId/3495

Martins, R., Caldeira, F., Sá, F., Abbasi, M., & Martins, P. (2020). An overview on how to develop a low-code application using OutSystems. *In Proceedings of the 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)* (pp. 395–401). IEEE. https://doi.org/10.1109/ICSTCEE49637.2020.9277404

Sattar, N. A. (2018). *Selection of low-code platforms based on organization and application type* [Master's thesis]. LUTPub. https://urn.fi/URN:NBN:-fi-fe2018080233306

Sufi, F. (2023). Algorithms in Low-Code-No-Code for Research Applications: A Practical Review. *Algorithms*, *16*(2), 108. https://doi.org/10.3390/a16020108

Thite, M., Kavanagh, M. J., & Johnson, R. D. (2012). Evolution of human resource management & human resource information systems: The role of information technology. In M. J. Kavanagh, M. Thite, & R. D. Johnson (Eds.), *Human resource information systems*: *Basics, applications, and future directions* (pp. 2–34). Thousand Oaks, CA: Sage Publications.