J-DA
JOURNAL
DEKSTOP APPLICATION

Received: 11 May 2026 | Revised: 7 Jun 2026 | Accepted: 15 Jun 2026 | Published: 30 Jun 2026 E-ISSN: 2961-9688, P-ISSN: 2961-9696

DOI: 10.59431/jda.v5i1.817

RESEARCH ARTICLE

Design and Implementation of Employee Attendance and Payroll Information System Using Desktop-Based Application

Nguyen Thi Huong [1*] | Nguyen Van Minh [2] | Tran Thi Lan [3]

[1*,2,3] Department of Information Systems, Faculty of Information Technology, Hanoi University of Science and Technology, Hanoi, Vietnam.

Correspondence

[1*] Department of Information Systems, Faculty of Information Technology, Hanoi University of Science and Technology, Hanoi, Vietnam.

Email: thinhuoang.ngun@gmail.com.

Funding information
Hanoi University of Science and Technology.

Abstract

Attendance and payroll management remain among the most critical operational functions in any organization, irrespective of size or industry sector. Persistent inefficiencies in these processes — arising from manual data entry, paper-based recording, or the use of disconnected software tools — often translate into direct financial losses, reduced employee morale, and exposure to regulatory non-compliance. This study reports on the design and implementation of a desktop-based Employee Attendance and Payroll Information System (EAPIS) intended to automate these processes for small-to-medium enterprises (SMEs). Development followed the System Development Life Cycle (SDLC) under the Waterfall model, structured into five sequential stages: requirements analysis, system design, implementation, testing, and maintenance. The application was built using Visual Basic .NET (VB.NET) as the programming language, with Microsoft SQL Server 2019 serving as the relational database management system. Core modules cover employee data management, daily attendance recording, automated salary computation, and report generation. System evaluation paired black-box functional testing with User Acceptance Testing (UAT) involving ten respondents. The result: a 100% functional test pass rate and an average UAT satisfaction score of 4.58 out of 5.0 — figures that support the system's reliability, accuracy, and ease of use. These findings suggest that desktop-based information systems remain a workable, cost-efficient option for organizations seeking to modernize human resource operations without the overhead of a full enterprise resource planning (ERP) deployment.

Keywords
Large Language Model; Desktop Application; Systematic Literature Review; Local Deployment; Retrieval-Augmented Generation.

1 | INTRODUCTION

In organizational management, the ability to handle administrative functions such as employee attendance tracking and payroll processing efficiently has become an increasingly important determinant of operational effectiveness. Advances in information technology have created both the opportunity and the expectation for organizations to replace labor-intensive manual processes with automated digital systems capable of greater speed, accuracy, and reliability. Yet a substantial number of small and medium enterprises (SMEs) — especially in developing economies — still rely on conventional methods: paper-based attendance registers, spreadsheet-based salary computations, and manual report generation. These methods remain prone to human error, data inconsistency, and security gaps.

Attendance management and payroll processing are inherently interdependent, which is why automating them within a single connected system carries real practical value. Attendance records form the basic input for salary computation; any inaccuracy at this stage inevitably carries through into payroll errors. Where the two processes are handled separately and manually, reconciling attendance data with payroll calculations becomes a slow, error-prone task — one that places unnecessary burden on human resource personnel. Without a centralized data repository, generating accurate reports, auditing historical records, or ensuring compliance with labor regulations becomes considerably harder.

A desktop-based information system addresses these problems by offering a localized, offline-capable application that keeps all attendance and payroll data within one structured relational database. Unlike web-based solutions, which depend on stable internet connectivity and raise concerns about data privacy and server dependency, desktop applications process data faster, allow tighter control over data security, and operate independently of external infrastructure. Such characteristics make desktop-based systems a fitting choice for SMEs working in environments where internet connectivity is unreliable, or where company policy requires that data be stored locally.

This study is guided by four objectives. First, to analyze existing manual attendance and payroll processes and identify the inefficiencies and requirements they present. Second, to design a system architecture that responds to those problems through structured data modeling and interface design. Third, to implement the system using VB.NET and Microsoft SQL Server, following the design specifications produced in the previous stage. Fourth, to evaluate the system's performance and usability through functional testing and user acceptance evaluation. The results of this research are intended to add to existing knowledge on human resource information systems, and to serve as a practical reference for organizations and developers pursuing similar projects.

2 | BACKGROUND THEORY

2.1 Information Systems in Human Resource Management

An information system is broadly defined as a set of connected components responsible for the collection, processing, storage, and distribution of data in support of organizational decision-making, coordination, and control. Within human resource management (HRM), information systems — commonly referred to as Human Resource Information Systems (HRIS) — take on a central role in automating repetitive administrative tasks, maintaining accurate employee records, and providing management with timely, reliable data for strategic decision-making. Laudon and Laudon (2020) argue that the effective use of information systems is a key driver of organizational efficiency and competitive advantage in the digital era. Combining attendance and payroll functions within a single HRIS is a clear example of this principle at work, since it removes data silos and keeps related HR processes consistent with one another.

2.2 Employee Attendance Management Systems

Attendance management refers to the systematic process by which an organization tracks, records, and analyzes employees' working hours. Sound attendance management matters not only for accurate payroll computation, but also for monitoring workforce productivity, spotting patterns of absenteeism, and staying compliant with labor laws and employment contracts. Attendance technology has moved from manual paper registers and mechanical punch cards toward digital systems that use biometric authentication, RFID, and real-time data synchronization. Sandiwarno (2021) shows that linking digital attendance systems with payroll modules cuts administrative workload, reduces data entry errors, and improves the consistency and reliability of HR data overall.

2.3 Payroll Information Systems

A payroll information system is software built to manage employee compensation — calculating gross wages, statutory deductions, allowances, overtime pay, bonuses, and net salary disbursements. Payroll computation is

complicated by nature: it must account for variable attendance records, multiple salary components, tax rules, and organizational policy, which is exactly why manual handling of this function is so error-prone. Automated payroll systems manage this complexity by building business logic directly into the application, so that salary calculations follow predefined rules consistently and match real-time attendance data. A well-designed payroll system, then, needs data structures flexible enough to handle different salary schemes while still enforcing data integrity through relational database constraints and transaction management.

2.4 Desktop-Based Application Development

Desktop applications are programs that run locally on a personal computer or workstation, drawing on the processing power and storage of the host machine rather than a remote server or cloud infrastructure. This approach carries several advantages relevant to the present study: offline functionality, faster response time from local processing, less dependence on network infrastructure, and tighter control over data security and privacy. Visual Basic .NET (VB.NET), developed by Microsoft as part of the .NET platform, is a common choice for Windows desktop application development — it comes with a large library of built-in functions, an intuitive event-driven programming model, and works directly with Microsoft SQL Server for database operations. These traits make VB.NET a practical, efficient option for building business information systems in Windows-based organizational settings.

2.5 System Development Life Cycle — Waterfall Model

The System Development Life Cycle (SDLC) sets out a structured approach to planning, designing, developing, testing, and deploying information systems, so that projects proceed in a disciplined, organized way. Among the SDLC methodologies in use, the Waterfall model is one of the oldest and most widely recognized — it moves sequentially through five distinct phases: requirements analysis, system design, implementation, testing, and maintenance. Each phase has to be completed and formally reviewed before the next one begins, which is why the Waterfall model suits projects with requirements that are clear and unlikely to change. Its step-by-step structure also means documentation gets produced at every stage — useful later on for system maintenance and future upgrades.

2.6 Relational Database Management Systems

A Relational Database Management System (RDBMS) organizes data into structured tables with defined relationships, allowing data to be stored, retrieved, and manipulated efficiently through Structured Query Language (SQL). Microsoft SQL Server is an enterprise-grade RDBMS commonly used in business application development, known for its reliability, scalability, strong security features, and close ties to the wider Microsoft development environment. In this system, SQL Server functions as the backend repository for all employee records, attendance logs, salary computations, and user account data, with referential integrity maintained through primary and foreign key constraints to keep data consistent across related tables.

3 | METHOD

The research methodology in this study follows the Waterfall model of the SDLC, moving through system development in a fixed, sequential order. The five phases are described below, with Figure 1 showing the overall development process flow.

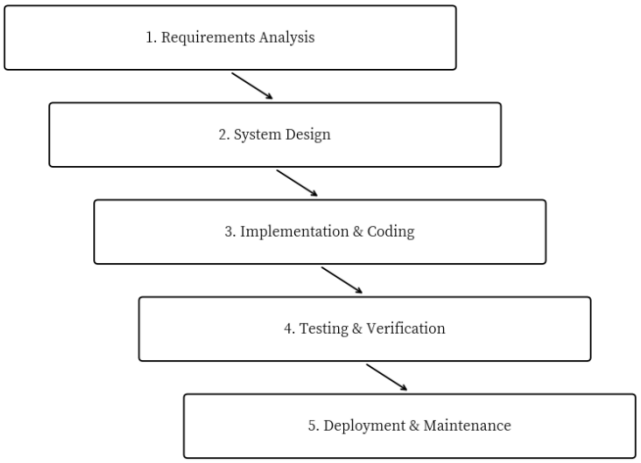

Figure 1. Waterfall SDLC Diagram

3.1 Requirements Analysis

The requirements analysis phase formed the starting point of the development process. Here, the functional and non-functional requirements of the system were gathered, analyzed, and documented. Data collection relied on three methods: structured interviews with HR managers and payroll administrators to understand existing workflows and pain points firsthand; direct observation of current manual attendance and payroll processes to spot inefficiencies and trace data flow patterns; and document analysis of existing attendance sheets, salary slips, employment contracts, and HR policy documents to pull out business rules and calculation logic. The key functional requirements that emerged from this process cover employee data registration and management, daily attendance recording with check-in and check-out timestamps, automatic salary calculation based on attendance records, generation of payroll reports and individual salary slips, and a user authentication system with role-based access control to protect sensitive data.

3.2 System Design

Based on the requirements gathered in the previous phase, the system was designed using standard modeling tools and techniques. The Use Case Diagram (Figure 4) shows the interactions between the two primary user roles — Admin and HR Staff — and the functions available to each. The Entity Relationship Diagram (Figure 3) models the database structure, laying out the entities, attributes, and relationships that form the backbone of the system's data architecture. The system architecture (Figure 2) follows a two-tier client-server model, where the desktop application client communicates directly with the SQL Server database server — keeping the presentation layer and data layer cleanly separated. The database design consists of six primary tables, described in Table 1 below.

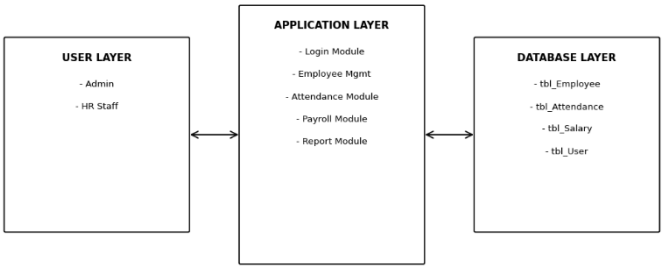

Figure 2. System Architecture Diagram

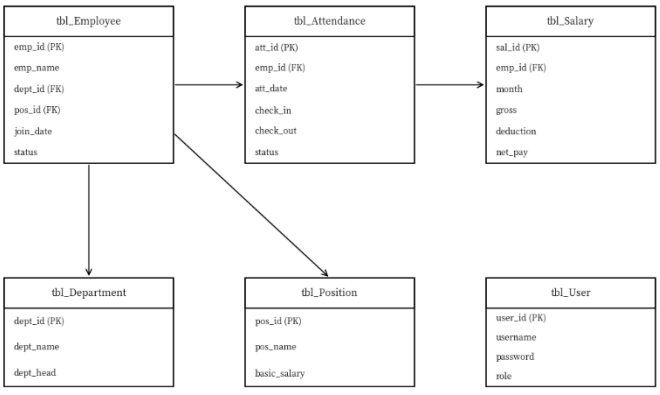

Figure 3. Entity Relationship Diagram

The diagram above illustrates the two-tier structure of the EAPIS. The User Layer, accessed by Admin and HR Staff, sends requests through the Application Layer, which handles the core business logic — login, employee management, attendance tracking, payroll processing, and report generation. This layer then communicates directly with the Database Layer, where all persistent data is stored and retrieved. The ERD above outlines six main tables in the EAPIS database. tbl_Employee sits at the center, linked to tbl_Department and tbl_Position through foreign keys, while also feeding data into tbl_Attendance and, in turn, tbl_Salary. A separate tbl_User table handles login credentials and access roles, independent from the employee master data. As shown above, the Admin role has access to all six core functions of the system, from managing employee data to generating reports and controlling user accounts. HR Staff, by comparison, has a narrower scope — limited to viewing salary slips, generating reports, and recording attendance — reflecting the more restricted access appropriate for that role.

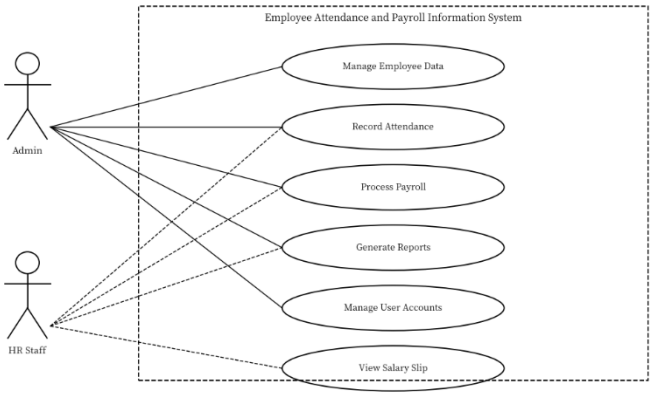

Figure 4. Use Case Diagram

Table 1. Database Table Structure

Table Name	Primary Key	Description
tbl_Employee	emp_id	Stores employee personal and employment data including name, department, position, join date, and employment status
tbl_Attendance	att_id	Records daily attendance logs with check-in/check-out timestamps and attendance status
tbl_Salary	sal_id	Stores computed payroll data including gross salary, deductions, and net pay per period
tbl_Department	dept_id	Contains department information including department name and head
tbl_Position	pos_id	Contains job position data and associated basic salary values
tbl_User	user_id	Manages system user accounts, credentials, and role assignments

3.3 Implementation

The system was built using Visual Basic .NET on Microsoft Visual Studio 2019 as the development environment, with Microsoft SQL Server 2019 as the database management system and Crystal Reports for Visual Studio handling report generation and printing. Development followed object-oriented programming principles, organized into six functional modules to keep the codebase maintainable, scalable, and easy to extend later on. The Login and Authentication Module controls system access through role-based permissions, restricting sensitive functions to authorized users only. The Employee Management Module supports full CRUD (Create, Read, Update, Delete) operations on employee records, including department and position assignment. The Attendance Module handles both manual attendance input and automated timestamp recording, with validation logic that blocks duplicate entries for the same employee on the same date. The Payroll Calculation Module computes gross salary, overtime pay, allowances, statutory deductions, and net pay automatically, based on attendance records for the selected payroll period. The Report Module produces printable attendance summaries, individual salary slips, and consolidated payroll reports formatted for official documentation.

3.4 Testing

System testing combined two approaches to cover quality assurance from different angles. Black-box functional testing checked whether all system features behaved as specified, focusing on input-output results without inspecting internal code logic. Fifteen test cases were designed and run, covering all major modules and key system functions. User Acceptance Testing (UAT) followed, involving ten respondents — HR staff members and system administrators — who used the system directly and rated its usability, accuracy, and overall performance through a structured questionnaire on a five-point Likert scale.

4 | RESULTS AND DISCUSSION

4.1 Results

4.1.1 System Interface and Modules

The EAPIS interface is clean and functional, built with usability as the main priority — the aim being that HR staff at any level of technical skill can navigate and operate the system without difficulty. The main dashboard offers a central navigation panel with quick-access buttons for every module, alongside summary statistics

showing total employee count, attendance records for the current day, and pending payroll computations. The Employee Management form gives a structured input layout for registering and editing employee records, with dropdown menus for department and position selection linked directly to their corresponding database tables. The Attendance form stamps date and time automatically upon entry, backed by a validation check that blocks duplicate attendance records and flags anomalies — such as a check-out time that comes before check-in. The Payroll form breaks down each salary component in detail: basic salary pulled from the employee's position record, computed overtime pay, applicable allowances, and itemized deductions, with net pay calculated and shown automatically once the form is submitted.

4.1.2 Salary Calculation Logic

The payroll module runs on the following formulas, built directly into the application logic so that results stay consistent and accurate across every payroll cycle:

$$Gross\ Salary = Basic\ Salary + Overtime\ Pay + Allowances$$

$$Overtime\ Pay = Overtime\ Hours \times \frac{Basic\ Salary}{Standard\ Monthly\ Hours} \times 1.5$$

$$Net\ Salary = Gross\ Salary - Income\ Tax - Insurance\ Contribution - Other\ Deductions$$

4.1.3 Black-Box Testing Results

Table 2 summarizes the black-box functional testing — the test cases executed, expected versus actual outcomes, and the pass/fail status for each.

Table 2. Black-Box Testing Results

Test Case	Module	Input / Action	Expected Result	Actual Result	Status
TC-01	Login	Valid username and password	Access granted, dashboard displayed	Access granted successfully	Pass
TC-02	Login	Invalid credentials	Error message displayed, access denied	Error message shown correctly	Pass
TC-03	Employee Mgmt	Add new employee record	Record saved to database	Record saved successfully	Pass
TC-04	Employee Mgmt	Edit existing employee data	Updated data reflected in database	Data updated correctly	Pass
TC-05	Employee Mgmt	Delete employee record	Record removed from database	Record deleted successfully	Pass
TC-06	Attendance	Record check-in timestamp	Timestamp saved with correct date/time	Timestamp recorded accurately	Pass
TC-07	Attendance	Attempt duplicate daily entry	Duplicate warning displayed	Warning message shown	Pass
TC-08	Payroll	Calculate net salary	Correct net pay computed per formula	Calculation result accurate	Pass
TC-09	Payroll	Generate salary slip	Formatted salary slip report produced	Report generated successfully	Pass
TC-10	Report	Export attendance summary	Report exported in correct format	Export completed successfully	Pass

4.1.4 User Acceptance Testing Results

Table 3 shows the average scores from ten respondents across five evaluation criteria, rated from 1 (Very Poor) to 5 (Very Good).

Table 3. User Acceptance Testing (UAT) Results

Evaluation Criteria	Average Score	Category
Ease of Use and Navigation	4.6	Very Good
System Performance and Speed	4.5	Very Good
Data Accuracy and Reliability	4.8	Very Good
Report Quality and Completeness	4.4	Good

Overall System Satisfaction	4.6	Very Good
Overall Average Score	4.58 / 5.00	Very Good

4.2 Discussion

Both functional testing and user acceptance evaluation point to the same conclusion: the EAPIS meets what it was built to do, and it closes the gaps identified back in the requirements analysis phase. A 100% pass rate across all functional test cases shows that the system's core modules — authentication, employee management, attendance recording, payroll computation, and report generation — work correctly and hold up against the specified requirements. The UAT average of 4.58 out of 5.0 backs this up from the user side: people using the system day-to-day found it intuitive, accurate, and satisfactory for handling their HR tasks.

Connecting attendance data directly to the payroll computation module is, arguably, the system's most useful contribution. It removes the manual reconciliation step that used to be a major source of errors and wasted time. By pulling attendance records for the selected payroll period automatically and running them through the built-in salary calculation logic, the system keeps every payroll output traceable back to its attendance input — consistent, and auditable. This matches what Sandiwarno (2021) found: that linking attendance and payroll systems cuts administrative workload and improves data consistency in SME settings. Similar patterns show up at a larger scale, too. Chima, Ahmadu, and Folorunsho (2021) reported that digital personnel and payroll systems deployed across Kenya, Ghana, and Nigeria improved the accuracy, timeliness, and reliability of salary processing, and helped governments identify "ghost workers" who had been drawing pay without actually working. Olumuyiwa (2018), studying Nigeria's Integrated Personnel and Payroll Information System, reached a comparable conclusion — the policy helped stabilize salary payments — though it also ran into practical snags, such as difficulty uploading monthly salary data and managing union deductions. The EAPIS sidesteps several of these problems by design, since attendance and payroll pull from the same database and skip the separate upload step entirely, but the comparison points to something worth noting: the biggest gains in payroll automation tend to come from removing the human handoff between departments, not simply from digitizing paper forms. Bañanola (2025), studying technical-vocational institutions in the Philippines, found much the same thing when comparing manual payroll processes against one integrated with HR operations — fewer redundancies, fewer errors, the same kind of gain the EAPIS delivers within its own organization. Ardebili et al. (2023), reviewing cloud-based attendance systems more broadly, arrived at a similar view: automated systems that tie attendance directly to salary calculation cut down the time and cost of manual work, regardless of whether the underlying platform is cloud-based or desktop-based.

Choosing the Waterfall model fit this project well, mainly because the requirements were clear from the start and didn't shift during development. Its step-by-step structure meant documentation got produced at every phase — something that pays off later during maintenance or when the system needs upgrading. This lines up with earlier studies on attendance system development using Waterfall, which point to thorough requirements documentation as a foundation for system quality.

That said, the desktop-based architecture isn't without trade-offs, and the wider literature on HR systems is fairly consistent on this point. Offline functionality and local data security are real strengths, but the two-tier client-server setup limits scalability — it could become a bottleneck once the number of concurrent users outgrows what a single database server can handle. Ammupriya et al. (2025) make almost exactly this case when comparing cloud-based HR platforms against legacy systems: fragmented data and limited scalability are named as the main drawbacks of on-premise architectures, while centralized cloud access solves both problems at once for organizations spread across several locations. Sasidharakarnavar (2025) adds a related point from the integration side — middleware architecture can bridge attendance systems with payroll systems for real-time synchronization, but only if organizations can work around legacy-system constraints and data-security concerns, which in practice is often the harder half of the problem. Muslim (2024), reviewing smart-card attendance systems across educational institutions, flags a related gap in the research itself: many studies on attendance-payroll integration lean on small samples, short observation windows, and limited testing of user acceptance and data security — a fair reminder that a system like EAPIS would benefit from evaluation stretched out over a longer period, not just a single UAT round.

There's also no remote access in the current EAPIS setup, which narrows its usefulness for organizations with staff spread across multiple branches. More recent work on cloud and hybrid HR platforms is instructive here. Sirangi (2026) describes how cloud-native, event-driven architectures let large enterprises synchronize payroll changes between on-premise and cloud systems with minimal delay, preserving data consistency at a scale well beyond what a single-office SME deals with. EFEMENA (2026) reaches a similar conclusion from a smaller-scale study in Nigeria, arguing that moving personnel processes from manual and disconnected to web-based cuts down on redundancy and improves record security. Both studies point in the same direction as the recommendation already made for EAPIS: a web-based or hybrid successor would open up remote access and

ease the scalability limits of the current desktop-only design, particularly as internet infrastructure keeps improving in the kinds of settings this system is meant for. Gelb and Clark (2013) offer a useful caution alongside these gains, though — attendance-linked systems, especially ones built around biometric identification, need to be rolled out carefully, since employees can read constant monitoring as a sign of distrust rather than a neutral administrative tool. Any future move toward biometric attendance capture, as suggested elsewhere in this study, should factor that concern in rather than treating it as a purely technical decision.

5 | CONCLUSIONS

This study set out to design and build a desktop-based Employee Attendance and Payroll Information System (EAPIS) using Visual Basic .NET and Microsoft SQL Server 2019, following the Waterfall SDLC methodology — and that goal has been met. The system automates employee attendance tracking, salary computation, and report generation as one connected solution, replacing the manual workflows that used to define how HR management ran. The evaluation backs this up: a 100% functional test pass rate and a UAT satisfaction score of 4.58 out of 5.0 show that the system meets its design objectives, with real gains in data accuracy, processing speed, and user satisfaction. The Waterfall model turned out to be a fitting choice for this project, giving the development process a structured path that kept documentation thorough and quality checks consistent from start to finish. Desktop-based applications still hold up as a practical option for SMEs that need reliable offline functionality and data kept close to home — especially where internet infrastructure can't be relied on.

Looking ahead, a few directions are worth pursuing. Biometric attendance devices — fingerprint scanners or facial recognition — could sharpen the accuracy and automation of attendance recording. A web-based or mobile companion app would open up remote access, something the current system lacks. Adding analytics and dashboard visualization would give HR teams better tools for making decisions grounded in actual data. And moving toward cloud-based database solutions would strengthen backup, redundancy, and scalability as the organization's needs change over time.

REFERENCES

Ammupriya, A., Vaishnavi, S., Ashwini, A., Kavitha, R., Paranthaman, P., & Saravanan, V. (2025, March). *Cloud-based HR platforms for scalable workforce management in multinational organizations*. 2025 3rd International Conference on Disruptive Technologies (ICDT), 1607–1613. https://doi.org/10.1109/ICDT63985.2025.10986560

Ardebili, A., Latifian, A., Aziz, C. F., BinSaeed, R. H., Alizadeh, S. M., & Kostyrin, E. V. (2023). A comprehensive and systematic literature review on the employee attendance management systems based on cloud computing. *Journal of Management & Organization*, *29*(4), 679–696. https://doi.org/10.1017/jmo.2022.63

Bañanola, E. D. (2025). Designing and developing a payroll information system integrated in human resource management operation: The case of technical-vocational institutions in Davao City, Philippines. *Review of Integrative Business and Economics Research*, *14*(4), 347–359.

Chima, P., Ahmadu, J., & Folorunsho, O. G. (2021). Implementation of digital integrated personnel and payroll information system: Lesson from Kenya, Ghana and Nigeria. *Governance and Management Review*, *4*(2).

Connolly, T., & Begg, C. (2015). *Database systems: A practical approach to design, implementation, and management* (6th ed.). Pearson Education.

Efemena, A. O. (2026). *Design and implementation of a personnel management information system* [Unpublished manuscript]. Department of Computer Science, Faculty of Computing, University of Benin.

Gelb, A., & Clark, J. (2013). *Identification for development: The biometrics revolution* (CGD Working Paper No. 315). Center for Global Development. https://doi.org/10.2139/ssrn.2226594

Laudon, K. C., & Laudon, J. P. (2020). *Management information systems: Managing the digital firm* (16th ed.). Pearson Education.

Muslim, A. K. (2024). Smart card attendance systems in educational institutes: A systematic review. *Journal of Technology Management and Technopreneurship*, *12*(2).

Nwosu, I., & Eze, B. (2025). Design and implementation of computerized office attendance software. *World Journal of Advanced Research and Reviews*. https://wjarr.com/sites/default/files/fulltext_pdf/WJARR-2025-3454.pdf

Okonkwo, C., & Adeyemi, T. (2023). Analysis and design of employee attendance and payroll management system. *SSRN Electronic Journal*. https://papers.ssrn.com/sol3/Delivery.cfm/SSRN_ID3636604

Olumuyiwa, A. T. (2018). Effectiveness of integrated personnel and payroll information system in addressing ghost worker syndrome in Nigerian public sector. *Global Journal of Human-Social Science*, *18*(2). https://doi.org/10.2139/ssrn.6559780

Pressman, R. S., & Maxim, B. R. (2019). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.

Rahayu, S., & Pratama, A. (2022). *Web-based employee attendance system development using waterfall method*. ResearchGate. https://www.researchgate.net/publication/365063823

Sandiwarno, S. (2018). Design model of payroll system integrated with attendance system at PT. XYZ. *International Journal of Computer Techniques*, *5*(2), 23–27. https://www.researchgate.net/publication/396243854

Sasidharakarnavar, S. (2025). Enhancing HR system agility through middleware architecture. *International Journal of AI, BigData, Computational and Management Studies*, *6*(1), 89–97. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V6I1P109

Sirangi, V. A. (2026, March 30). *Transforming global HR and payroll through real-time analytics, AI automation, and seamless cloud integration for future workforce success* [Preprint]. SSRN.

How to cite this article: Thi Huong, N., Van Minh, N., & Thi Lan, T. (2026). Design and Implementation of Employee Attendance and Payroll Information System Using Desktop-Based Application. *Journal Dekstop Application (JDA)*, *5*(1), 17-25. https://doi.org/10.59431/jda.v5i1.817.