

Design and Development of a Web-Based Employee Leave Management Information System at PT. Sekar Laut, Tbk

Moh. Robbi Abinadji¹, Arda Surya Editya^{1*}

¹Department of Informatics, Nahdlatul Ulama University of Sidoarjo, Indonesia.

Abstract

Information systems that specifically manage personnel data require special attention, particularly due to their role in ensuring efficiency and effectiveness in processing employee-related data such as employee data entry, data editing, data retrieval, and especially employee leave management. The existing employee leave management system at PT. Sekar Laut, Tbk is still not optimal. Employees are required to fill out manual leave request forms and wait for approval from supervisors, while inaccuracies in employee leave balances frequently occur.

To address these issues, a more effective employee information system is required to provide comprehensive information regarding employee leave and to simplify the process of identifying leave usage and remaining leave balances.

This study employs the waterfall development method, while system design is implemented using Unified Modeling Language (UML) diagrams. The expected outcome is a web-based employee leave information system capable of resolving existing system problems, optimizing the submission and approval process of employee leave, generating leave reports, and facilitating easier retrieval of employee leave data. The results of this study are expected to support PT. Sekar Laut, Tbk, particularly the Human Resources division, in developing an effective employee leave management information system.

Keywords

Employee Leave; Information Systems; UML Diagram; Waterfall Method; Web-Based System.

*Correspondence

Arda Surya Editya
ardasurya.tif@unusida.ac.id

Introduction

The rapid advancement of information technology has significantly transformed organizational business processes, particularly in the management of human resources. Information systems are no longer merely supportive tools but have become strategic components that enhance efficiency, accuracy, and decision-making within organizations. A well-designed information system enables organizations to manage data systematically, reduce human error, improve data security, and accelerate access to critical information (Laudon et al, 2020), (Kurniawan H, 2021). In the context of modern enterprises, the integration of web-based information systems has become essential to support operational effectiveness and organizational competitiveness (Nugraha et al, 2017).

Human resource management is one of the organizational domains that requires precise and reliable information systems. Personnel information systems facilitate the management of employee data, including recruitment records, attendance, payroll, and leave administration. Employee leave management, in particular, plays a crucial role in maintaining work–life balance, employee satisfaction, and compliance with labor regulations. However, many organizations still rely on manual or semi-manual leave management processes, which often result in data inconsistencies, delayed approvals, and inaccuracies in leave balance calculations (Hutahaeen J, 2015), (Pratama E. B., 2020).

At PT. Sekar Laut, Tbk, the employee leave application process is still conducted using paper-based forms that require manual submission and approval by supervisors and the Human Resources (HR) department. This approach introduces several operational challenges, such as prolonged approval cycles, difficulties in tracking leave history, and frequent discrepancies between recorded and actual leave balances. These limitations highlight the need for a computerized employee leave information system that can automate workflows, ensure data integrity, and provide real-time access to leave information for both employees and management.

Previous studies have demonstrated that web-based personnel information systems can significantly improve efficiency, transparency, and accuracy in administrative processes. Research on web-based employee leave systems using the Waterfall development model shows that structured system development contributes to clearer requirement definition and more reliable system implementation (Kurniawan H, 2021), (Pratama E. B., 2020), (Tabrani et al, 2019). Furthermore, the use of Unified Modeling Language (UML) has been widely adopted to model system requirements and architectures in an object-oriented manner, enabling better communication between developers and stakeholders (Dennis et al, 2020), (Rosa et al, 2018).

The Waterfall method remains a relevant and widely applied software development approach for information systems with clearly defined requirements and stable business processes. Its sequential phases—analysis, design, implementation, testing, and maintenance—provide systematic guidance for developing robust and maintainable systems (Pressman et al, 2020), (Sommerville I, 2016). When combined with UML-based modeling, the Waterfall method offers a comprehensive framework for designing enterprise-scale information systems, including human resource management applications.

Based on these considerations, this study focuses on the design and development of a web-based employee leave management information system for PT. Sekar Laut, Tbk. The system is expected to optimize the leave application and approval process, provide accurate leave balance calculations, and generate comprehensive leave reports. By implementing a structured development methodology and established software engineering principles, this research aims to contribute both practically—by improving HR operational processes—and academically—by enriching studies related to personnel information systems and software engineering practices.

Methodology

This study adopts a software development research approach with a focus on designing and implementing a web-based employee leave management information system. The selected development methodology is the Waterfall model, which is suitable for systems with well-defined requirements and structured business processes as shown in Figure 1. The Waterfall model emphasizes a sequential and systematic flow of development stages, ensuring that each phase is completed and validated before proceeding to the next stage (Pressman et al, 2020), (Sommerville I, 2016). This approach provides clarity in system planning, minimizes development risks, and supports thorough documentation throughout the software lifecycle.

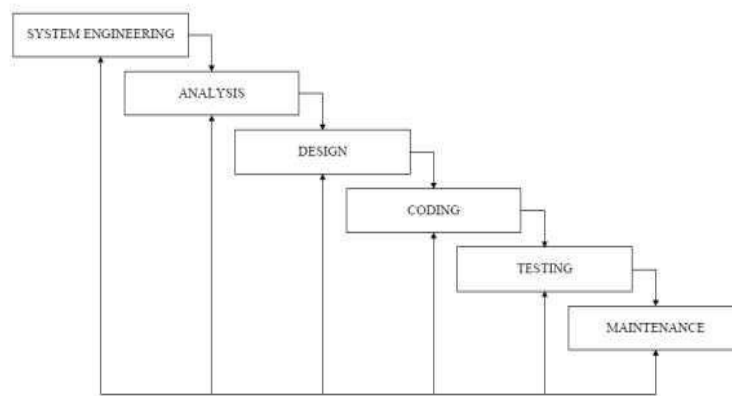


Figure 1. Waterfall Diagram

System Engineering and Requirement Analysis

The first phase of this research is System Engineering, which aims to understand the organizational context and operational needs of PT. Sekar Laut, Tbk. At this stage, data collection was conducted through direct observation of existing leave management procedures, informal interviews with employees and Human Resources (HR) staff, and analysis of organizational documents related to employee leave policies. In addition, relevant literature from books and scientific journals was reviewed to establish a theoretical foundation for personnel information systems and leave management applications (Laudon et al, 2020), (Nugraha et al, 2017).

Following system engineering, the research proceeds to the Requirement Analysis phase. This stage focuses on identifying functional and non-functional requirements of the proposed system. Functional requirements include employee leave submission, approval workflows, leave balance calculation, reporting, and user role management. Non-functional requirements encompass system usability, data security, reliability, and accessibility through web browsers. Clear documentation of user requirements is critical, as it serves as the primary reference for subsequent design and development phases and reduces ambiguity during system implementation (Dennis et al, 2020), (Rosa et al, 2018).

System Design and Implementation

The System Design phase translates the analyzed requirements into a structured system blueprint. In this research, system modeling is performed using Unified Modeling Language (UML), which is widely recognized for object-oriented system design. UML diagrams such as use case diagrams, activity diagrams, and class diagrams are utilized to represent system functionality, user interactions, data structures, and system workflows. The use of UML enhances communication between developers and stakeholders and ensures that system logic is clearly defined prior to coding (Dennis et al, 2020), (Rosa et al, 2018).

After the design phase, the research advances to System Implementation (Coding). At this stage, the approved system design is translated into a functional web-based application. The implementation process follows the predefined architecture and design specifications to maintain consistency and software quality. Coding activities focus on developing system modules for user authentication, leave submission, approval processing, leave balance management, and report generation. This structured implementation approach aligns with best practices in software engineering, ensuring maintainability and scalability of the developed system (Pressman et al, 2020), (Sommerville I, 2016).

Testing, Deployment, and Maintenance

Once the system implementation is completed, Testing is conducted to verify that the application meets all specified requirements. Functional testing is performed to ensure that each feature operates correctly, while integration testing validates the interaction between system modules. Black-box testing techniques are applied to evaluate system functionality from the user's perspective without examining internal code structures, focusing on input-output behavior and system responsiveness (Cholifah et al, 2018). This testing phase is essential to detect errors, ensure system reliability, and improve overall software quality.

The final phase of the methodology is Maintenance, which begins after the system is deployed and used by end users. Maintenance activities include correcting unforeseen errors, adapting the system to organizational changes, and enhancing system performance based on user feedback. Continuous maintenance ensures that the employee leave information system remains effective, secure, and aligned with organizational needs over time. This phase reflects the long-term perspective of software engineering, emphasizing sustainability and continuous improvement of information systems (Pressman et al, 2020), (Sommerville I, 2016).

Database Design

The database is designed to ensure **data integrity**, **traceability**, and **role-based access** for the employee leave workflow. A relational schema is selected because it supports structured HR data storage, consistent transactions, and reporting needs in organizational information systems (Laudon et al, 2020), (Nugraha et al, 2017). The schema centers on employee master data, leave types, leave balances, and leave applications, enabling the system to compute balances accurately and generate periodic reports.

Table 1. Users Table

Field Name	Data Type	Description
user_id (PK)	INT	Unique identifier for system users
username	VARCHAR	Login username (unique)
password_hash	VARCHAR	Encrypted user password
role	ENUM	User role (admin, hrd, manager)
name	VARCHAR	Full name of the user
email	VARCHAR	User email address
created_at	DATETIME	Account creation timestamp

Table 2. Employee Table

Field Name	Data Type	Description
employee_id (PK)	INT	Unique employee identifier
employee_code	VARCHAR	Employee identification number
full_name	VARCHAR	Employee full name
department	VARCHAR	Department name
position	VARCHAR	Job position
join_date	DATE	Employment start date
status	ENUM	Employment status (active/inactive)

Table 3. leave_types

Field Name	Data Type	Description
leave_type_id (PK)	INT	Unique leave type identifier
type_name	VARCHAR	Type of leave (annual, sick, maternity, etc.)
max_days_per_year	INT	Maximum leave days per year
description	TEXT	Description of leave type

Table 4. leave_balance

Field Name	Data Type	Description
balance_id (PK)	INT	Unique balance record identifier
employee_id (FK)	INT	Reference to employees.employee_id
leave_type_id (FK)	INT	Reference to leave_types.leave_type_id
year	INT	Leave year
entitlement_days	INT	Total entitled leave days
used_days	INT	Used leave days
remaining_days	INT	Remaining leave days

Table 5. leave_requests

Field Name	Data Type	Description
request_id (PK)	INT	Unique leave request identifier
employee_id (FK)	INT	Employee submitting leave
leave_type_id (FK)	INT	Type of leave requested
start_date	DATE	Leave start date
end_date	DATE	Leave end date
total_days	INT	Total leave days requested
reason	TEXT	Reason for leave
status	ENUM	Request status (submitted, approved, rejected, cancelled)
submitted_at	DATETIME	Submission timestamp

Table 6. leave_approvals

Field Name	Data Type	Description
approval_id (PK)	INT	Unique approval record identifier
request_id (FK)	INT	Reference to leave_requests.request_id
approver_user_id (FK)	INT	Approving user identifier
decision	ENUM	Approval decision (approved/rejected)
notes	TEXT	Approval notes or comments
decided_at	DATETIME	Decision timestamp

UML Design

UML is used to model system behavior and structure in an object-oriented manner, improving clarity of requirements, design communication, and implementation consistency (Dennis et al, 2020). The UML artifacts developed in this study include Use Case Diagram, Activity Diagram, and Class Diagram to represent functional scope, workflow logic, and system structure.

Use Case Model

Actor

- Employee: submits leave request, views leave balance/history, changes password.
- Manager: Review Leave Request, Approve/Reject Request, View Team Leave Report
- HR/Admin: Manage Employee Data, Manage Leave Types, Manage Leave Balances/Entitlements, Generate Reports, Manage Users/Roles

Main Use Case

- Employee: Login, Submit Leave Request, View Leave Balance, View Request Status/History, Change Password
- Manager: Review Leave Request, Approve/Reject Request, View Team Leave Report
- HR/Admin: Manage Employee Data, Manage Leave Types, Manage Leave Balances/Entitlements, Generate Reports, Manage Users/Roles

This structure ensures separation of responsibilities and aligns with role-based operational processes in HR information systems (Laudon et al, 2020), (Nugraha et al, 2017).

Activity Diagram

The activity flow for leave submission and approval is modeled to capture process logic end-to-end:

- 1) Employee logs in → opens “Apply Leave”.
- 2) Employee selects leave type, dates, and inputs reason.
- 3) System validates input (date range, eligibility, remaining balance).
- 4) If validation fails → system returns error and request is not stored.
- 5) If valid → request saved with status submitted.
- 6) Manager reviews request → decision approve/reject.
- 7) If approved → system updates leave balance (used/remaining) and sets status approved.
- 8) If rejected → status set rejected, balance unchanged.
- 9) Employee receives status update and can view history/report.

Workflow modeling like this reduces ambiguity in implementation and supports systematic development phases (Pressman et al, 2020), (Sommerville I, 2016).

Class Diagram

A simplified class model for implementation is defined as:

- a) User: userId, username, passwordHash, role, name
- b) Employee: employeeId, employeeCode, fullName, department, position
- c) LeaveType: leaveTypeId, typeName, maxDaysPerYear
- d) LeaveBalance: balanceId, year, entitlementDays, usedDays, remainingDays
- e) LeaveRequest: requestId, startDate, endDate, totalDays, reason, status
- f) LeaveApproval: approvalId, decision, notes, decidedAt

Associations

- a) Employee *has many* LeaveRequest
- b) LeaveRequest *has many* LeaveApproval
- c) Employee *has many* LeaveBalance
- d) LeaveRequest *references one* LeaveType
- e) LeaveApproval *references one* User (approver)

This object structure supports maintainable implementation and consistent mapping from analysis/design to code (Dennis et al, 2020), (Pressman et al, 2020), (Sommerville I, 2016).

Results and Discussion

Results

Black-Box Testing Results

Black-box testing was conducted to validate the system’s functional behavior by checking whether each feature produces the expected output for a given input, without evaluating the internal program code. The testing focuses on core workflows: authentication, leave submission, approval, leave balance update, and reporting.

Table 7. Black-Box Testing of the Web-Based Employee Leave System

Test ID	Module	Test Scenario	Test Data (Input)	Expected Output	Observed Output	Status
BB-01	Login (Employee)	Login with valid credentials	username/pass word valid	Redirect to employee dashboard	Dashboard displayed	Pass

Test ID	Module	Test Scenario	Test Data (Input)	Expected Output	Observed Output	Status
BB-02	Login (Employee)	Login with invalid password	valid username, wrong password	Error message shown, stay on login page	Error displayed	Pass
BB-03	Login (Admin/HR)	Admin login success	valid admin account	Redirect to admin dashboard	Dashboard displayed	Pass
BB-04	Leave Submission	Submit leave with valid dates	start $\leq$ end, reason filled	Request saved, status = submitted	Request stored & listed	Pass
BB-05	Leave Submission	Submit leave with end date before start date	start $>$ end	Validation error, request not saved	Validation shown	Pass
BB-06	Leave Submission	Submit leave exceeding remaining balance	total_days $>$ remaining_days	Rejected by validation	Validation shown	Pass
BB-07	Leave Submission	Submit leave with empty reason	reason = empty	Validation error	Validation shown	Pass
BB-08	Approval (Manager)	Approve submitted request	status=submitted $\rightarrow$ approve	status becomes approved	Status updated	Pass
BB-09	Approval (Manager)	Reject submitted request	status=submitted $\rightarrow$ reject	status becomes rejected; balance unchanged	Status updated	Pass
BB-10	Leave Balance	Balance update after approval	approved request total_days=N	used_days += N, remaining_days -= N	Balance updated	Pass
BB-11	Leave Balance	Balance unchanged after rejection	rejected request	no change in balance	No change	Pass
BB-12	Reporting	Generate leave report by period	date range selected	Report lists requests in range	Report displayed	Pass
BB-13	Security/Access	Employee tries to open admin page	employee session	Access denied / redirected	Blocked/redirected	Pass
BB-14	Change Password	Change password with correct old password	old=correct, new=valid	Password updated, re-login works	Updated & verified	Pass

The test results indicate that critical workflows—login validation, leave request submission, approval decision, balance calculation, and report generation—behave as expected. Input validation prevents common errors (invalid date range, missing reason, and exceeding leave balance), while role-based access control restricts unauthorized access to administrative modules. These results strengthen the evidence that the proposed system improves reliability compared with manual procedures, particularly in preventing leave-balance inconsistencies and accelerating approval cycles.

User Interface Results

The developed web-based employee leave information system provides a simple and user-friendly interface to support both employees and administrators in performing their respective tasks. The interface design emphasizes clarity, ease of navigation, and role-based access to system features.

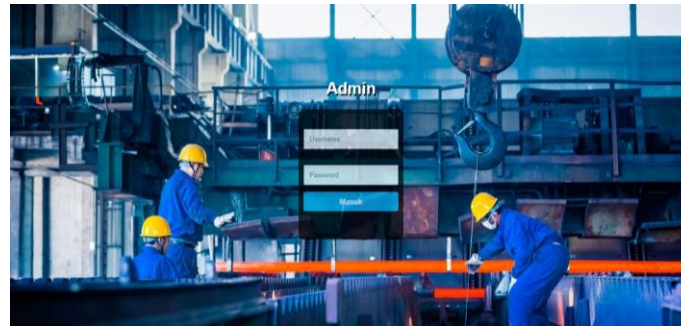


Figure 1. Login Page

The login interface ensures secure access by validating user credentials and directing users to dashboards based on their roles (employee, manager, or administrator). This role-based redirection prevents unauthorized access to system modules and supports secure system operation.

The employee dashboard allows users to submit leave requests, view leave balances, and monitor the status of submitted requests. The leave application form includes input validation to prevent incorrect data entry, such as invalid date ranges or requests exceeding available leave balances. This interface helps reduce user errors and improves the efficiency of the leave submission process.



Figure 2. Dashboard Employee Page

The administrator and manager interfaces provide access to employee data management, leave request review, and approval functions. Through these interfaces, decision-makers can efficiently process leave applications and generate leave reports without relying on manual documentation. Overall, the UI design supports effective system interaction while maintaining simplicity and usability for daily operational use.



Figure 3. Administrator Dashboard Page

Discussion

The implementation of the web-based employee leave management information system at PT. Sekar Laut, Tbk demonstrates a significant improvement in the efficiency and reliability of leave administration processes. Based on the black-box testing results, all core system functionalities operated as expected, indicating that the developed system successfully meets its functional requirements. This finding aligns with prior studies stating that computerized personnel information systems reduce operational errors and enhance administrative efficiency compared to manual systems (Laudon et al, 2020), (Pratama E. B., 2020).

One of the most notable improvements achieved by the proposed system is the automation of leave balance calculation. In the manual process, leave balance discrepancies frequently occurred due to delayed updates and human error. Through systematic validation and transactional updates, the system ensures that leave balances are updated only when leave requests are approved. This mechanism improves data accuracy and supports reliable decision-making by Human Resources and management, consistent with principles of information system reliability discussed in management information systems literature (Nugraha et al, 2017).

The adoption of the Waterfall development method contributes to the structured and systematic realization of the system. Clearly defined phases—from requirement analysis to testing—helped minimize requirement ambiguity and implementation errors. This result supports previous findings that the Waterfall model remains effective for organizational information systems with stable and well-defined business processes (Pressman et al, 2020), (Sommerville I, 2016), (Tabrani et al, 2019). Furthermore, UML-based modeling facilitated clearer communication between developers and stakeholders, ensuring that system functionality aligned with organizational needs (Dennis et al, 2020), (Rosa et al, 2018).

From a usability perspective, the simple and role-based user interface enables employees, managers, and administrators to interact with the system efficiently. Employees can submit leave requests and monitor their status in real time, while managers and HR staff can process approvals and generate reports without additional administrative overhead. These outcomes confirm that well-designed user interfaces enhance system acceptance and operational effectiveness in web-based information systems (Fridayanthie et al, 2016), (Hidayat R, 2017).

Overall, the results indicate that the developed employee leave information system not only addresses the limitations of the previous manual process but also provides a scalable and maintainable solution for future organizational needs. By integrating structured software engineering practices with practical HR requirements, the system offers both operational value and academic relevance in the domain of personnel information systems.

Conclusion

This study has successfully designed and implemented a web-based employee leave management information system for PT. Sekar Laut, Tbk using the Waterfall software development method and UML-based system modeling. The developed system addresses key limitations of the previous manual leave management process, particularly in terms of approval delays, data inconsistencies, and inaccuracies in leave balance calculation.

The results of functional validation through black-box testing demonstrate that all core system features—user authentication, leave submission, approval workflows, leave balance updates, and report generation—operate as expected. The automated validation mechanisms effectively prevent

common input errors and ensure that leave balances are updated accurately only after approval, thereby improving data reliability and administrative control.

From an operational perspective, the system enhances efficiency and transparency in human resource management. Employees are able to submit and monitor leave requests in real time, while managers and HR administrators can process approvals and generate reports more quickly and systematically. The use of a structured development methodology also contributes to system maintainability and scalability, making the application suitable for long-term organizational use.

In conclusion, the proposed employee leave information system provides a practical and effective solution for improving leave administration processes. This research contributes to the application of software engineering principles in personnel information systems and can serve as a reference for future studies or system development initiatives in similar organizational contexts.

References

- Cholifah, W. N., Yulianingsih, Y. and Sagita, S. M. (2018) 'Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap', *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, 3(2), pp. 206–210.
- Dennis, A., Wixom, B. H. and Tegarden, D. (2020) *Systems Analysis and Design: An Object-Oriented Approach with UML*. 6th edn. Hoboken, NJ: John Wiley & Sons.
- Fridayanthie, E. W. and Mahdiati, T. (2016) 'Rancang Bangun Sistem Informasi Permintaan ATK Berbasis Intranet (Studi Kasus: Kejaksaan Negeri Rangkasbitung)', *Jurnal Khatulistiwa Informatika*, 4(2), pp. 126–138.
- Hidayat, R. (2017) 'Aplikasi Penjualan Jam Tangan Secara Online', *Jurnal Teknik Komputer*, 3(2), pp. 90–96.
- Hutahaean, J. (2015) *Konsep Sistem Informasi*. Yogyakarta: Deepublish.
- Laudon, K. C. and Laudon, J. P. (2020) *Management Information Systems: Managing the Digital Firm*. 16th edn. Harlow: Pearson Education.
- Kurniawan, H. (2021) 'Perancangan Sistem Informasi Cuti Karyawan Berbasis Web Menggunakan Metode Waterfall', *Jurnal Informatika dan Rekayasa Perangkat Lunak*, 3(1), pp. 45–54.
- Nugraha, A. R. and Pramuka, G. (2017) 'Sistem Informasi Akademik Sekolah Berbasis Web di Sekolah Menengah Pertama Negeri 11 Tasikmalaya', *Jurnal Manajemen Informatika (JUMIKA)*, 4(2).
- Pratama, E. B. (2020) 'Penerapan Metode Waterfall dalam Perancangan Sistem Informasi Penggajian Karyawan', *Jurnal Informatika Universitas Pamulang*, 5(2), pp. 120–126.
- Pressman, R. S. and Maxim, B. R. (2020) *Software Engineering: A Practitioner's Approach*. 9th edn. New York: McGraw-Hill Education.
- Rosa, A. S. and Shalahuddin, M. (2018) *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.
- Sommerville, I. (2016) *Software Engineering*. 10th edn. Boston: Pearson.
- Tabrani, M. and Aghniya, I. R. (2019) 'Implementasi Metode Waterfall pada Program Simpan Pinjam Koperasi Subur Jaya Mandiri Subang', *Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi dan Komunikasi*, 14(1), pp. 44–53.