

Development of a Web-Based Daily Cash Flow Management System for Operational Finance at Riatar Playground Using the Agile Development Method

Amin Nur Ikhsanuddin ¹, Megawaty ^{2*}

^{1,2*} Department of Information Systems, Faculty of Science and Technology, Universitas Bina Darma, Palembang City, South Sumatra Province, Indonesia.

Article History:

Received: July 7, 2026.

Revised: July 10, 2026.

Accepted: August 27, 2026.

Available online: September 20, 2026.

Published: December 1, 2026.

*Corresponding Author:

megawaty@binadarma.ac.id

Abstract: Daily operational financial management is critical to the smooth running of business activities, particularly in the entertainment services sector, such as playgrounds. Riatar Playground handles a high volume of daily transactions, including ticket sales, play-area rentals, and registration fees for competitions, classes, and private lessons, making manual financial record-keeping prone to errors, delayed reporting, and difficulties in monitoring cash inflows and outflows. This study therefore aims to design and build a web-based daily cash flow system that supports operational financial recording in a structured, accurate, and easily accessible manner. The system development method used is Agile Development, which allows the process to be carried out in stages and adaptively according to user needs. The planning, design, development, and testing phases were carried out iteratively across five sprints based on direct feedback from Riatar Playground's owner and staff. The developed system features daily income and expense recording, transaction categorization, role-based access for Admin and Owner, and real-time presentation of cash flow reports through an interactive dashboard. System functionality was verified using black-box testing across seven scenarios covering authentication, transaction recording, report filtering, report export, and logout, all of which produced valid outputs, yielding a 100% test pass rate. The results demonstrate that the system enhances the effectiveness and efficiency of operational financial record-keeping, reduces recording errors, and enables managers to monitor daily financial conditions in real time, supporting more orderly, transparent, and accountable financial management for small and medium-sized service businesses with heterogeneous revenue streams.

Keywords: Daily Cash Flow; Operational Finance; Web-Based System; Agile Development; Black-Box Testing.

1. Introduction

Advances in information technology have fundamentally changed how organizations manage daily operations, and the adoption of web-based systems has become one of the most visible outcomes of this transformation. Across various industries, manual and paper-based procedures are gradually being replaced by digital platforms that offer faster processing, accessible data, and reliable documentation. Financial management benefits significantly from this shift, since the accuracy, timeliness, and traceability of records directly influence business sustainability. Small and medium-sized enterprises in the service and entertainment sectors increasingly recognize that a dedicated information system can substantially improve how financial activities are recorded, monitored, and evaluated. Children's entertainment venues, such as playgrounds, represent a service industry that continues to expand alongside public demand for recreational and educational activities. These businesses typically generate a high frequency of daily transactions, ranging from ticket sales and play-area rentals to registration fees for supplementary programs. Because operational cash flow fluctuates daily, financial records must be maintained systematically and accurately to facilitate real-time monitoring. Without an adequate recording mechanism, business owners and managers face difficulties in maintaining an up-to-date overview of their financial position, which can impede timely managerial decisions. Riatar Playground is a children's recreational facility that offers various daily activities, including entrance ticketing, play-area access, private English lessons, cooking classes, coloring competitions, and craft-making sessions. Each activity requires participant registration, processed either through direct cash payments or bank transfers. Although these offerings attract regular participants, the associated financial records remain unintegrated. Financial data is recorded across disparate handwritten notes, paper receipts, and informal spreadsheets, preventing management from obtaining a consolidated view of operational finances. This fragmented recording process introduces several operational challenges.

The absence of a centralized, web-based daily cash flow system leaves transactions prone to human errors, including arithmetic miscalculations, duplicate entries, and omitted records. Reconciling daily cash inflows and outflows is also time-consuming because staff must manually aggregate data from multiple media before producing summaries. Consequently, financial reporting is frequently delayed, and undocumented transactions pose financial risks that undermine accurate budgeting, cash flow tracking, and long-term planning. The operational complexity is further compounded by the absence of automated cross-verification between administrative staff handling front-desk transactions and business owners overseeing strategic finance. While prior technological implementations at Riatar Playground addressed customer-facing operations through a web-based e-ticket booking system (Ikhsanuddin & Megawaty, 2026), back-office financial recording remained largely unaddressed. As revenue streams broadened into educational workshops and skill-based sessions, cash handling became increasingly distributed among multiple staff members. Without a synchronized platform enforcing role-based transaction logging and systematic ledger closing, discrepancies between physical cash balances and recorded figures became difficult to isolate, creating administrative friction and increasing internal operational vulnerability. To address these issues, this study aims to design and develop a web-based daily cash flow system for Riatar Playground using the Agile development method. Agile was selected because its iterative and adaptive lifecycle accommodates evolving operational requirements through regular stakeholder feedback. Rather than following a rigid plan, the development team can evaluate functional prototypes at each stage and implement incremental refinements.

This approach aligns system features with the actual workflows of Riatar Playground and minimizes architectural discrepancies before final deployment. The resulting centralized system enables staff to record transactions systematically, provides real-time access to operational balances, and eliminates dependency on manual records. The scope of this research is centered on tracking cash inflows and outflows derived directly from playground operations, specifically entrance ticketing, activity registrations, class fees, operational expenses, and material purchases. Long-term corporate financing and non-operational capital investments remain outside the scope of this work. System evaluation is also bounded to functional black-box testing to verify that core transactional features perform accurately in accordance with user requirements, without addressing large-scale concurrent stress testing or advanced network security auditing. From a structural perspective, implementing a computerized ledger system with segregated user roles represents an essential mechanism for establishing operational transparency in recreational service enterprises. By formalizing transaction boundaries between Admin and Owner accounts, the system not only standardizes daily ledger closing but also facilitates immediate accountability for recorded receipts and disbursements. Consequently, digital operational cash recording serves as a foundational step toward data-driven administrative management, allowing small enterprise owners to mitigate transaction leakage and identify seasonal operational expenditure patterns. The system is intended to provide both practical and operational benefits. For Riatar Playground, digitized recording minimizes manual accounting errors, accelerates transaction reconciliation, and provides actionable financial summaries through an interactive dashboard. For the broader field of applied software engineering, this research provides an empirical reference on implementing Agile

principles to build tailored financial information systems for small-scale recreational service businesses managing diversified revenue streams.

2. Related Work

Thirteen studies pertinent to the development of web-based financial information systems published between 2021 and 2026 are reviewed in this section, with approximately 69% published within the last five years. To determine the strengths, limitations, and positioning of the current research relative to existing literature, the reviewed works are arranged thematically by development strategy and subsequently compared.

2.1 State-of-the-Art Research

Recent research on financial information systems can be classified into three primary streams based on development methodology. The first stream comprises Agile-based approaches. For Yayasan Umar Faruq Al Arifin, Zulfikar and Qoiriah (2025) utilized an Agile Kanban framework to transition a manual financial workflow into a web-based system, reporting improved operational transparency and structured task monitoring via digital Kanban boards. Focusing on streamlined data input, Andika Isma *et al.* (2024) developed a web-based financial system for CV Speed Computer that enabled owners to monitor financial conditions remotely and in real time. Fauzan Islamil Haq *et al.* (2024) demonstrated that intensive observation and requirements gathering within an Agile lifecycle produced a community-level (RT) financial system that satisfied operational needs, verified through functional black-box testing. In an investigation directly related to the current research venue, Ikhsanuddin and Megawaty (2026) developed a web-based e-ticket booking application for PT Playground Riatar using an Agile Scrum framework, demonstrating that transaction digitalization reduced ticketing queues and improved service delivery. In another Scrum-based application, Sabila *et al.* (2021) developed a financial reporting application for student organizations, showing that sprint-based incremental delivery effectively converted manual administrative records into digital ledgers, although its scope remained limited to basic organizational accounting.

The second stream employs sequential or prototyping-oriented development methodologies. To minimize manual calculation errors and accelerate financial report generation, Riska Aryanti *et al.* (2022) applied the Waterfall model to construct a project financial information system for a construction enterprise. Similarly, Ashifuddin and Neni Lisa Firana (2025) implemented the Waterfall model to design a smart financial platform for an early-childhood educational institution, incorporating automated ledger calculations and an administrative dashboard. Prototyping methodologies were adopted by Rahmatika and Edi (2022) and Nugroho (2022) to develop financial systems for village administrations and local treasuries, incorporating role-based access control to distinguish administrative data entry from community oversight. The third stream focuses primarily on architectural system modeling and empirical quality evaluation rather than specific development lifecycles. Utami Dewi and Voutama (2024) and Djohan *et al.* (2025) utilized Unified Modeling Language (UML) artifacts—specifically use case, activity, sequence, and class diagrams—to specify system requirements and align architectural comprehension between developers and end users. In an evaluative study of the village financial system (Siskeudes), Maulana *et al.* (2025) observed that data completeness and reporting transparency substantially determine information system effectiveness, while infrastructure and human resource readiness present ongoing challenges. Evaluating software quality via the ISO/IEC 25010 standard, Anwar *et al.* (2026) assessed an enterprise financial platform, reporting high compliance in functional suitability (84.29%) and usability (85.13%).

2.2 Comparison with Previous Studies

Table 1 summarizes the evaluated studies, comparing their development methodologies, case study domains, validation methods, and principal outcomes or constraints.

Table 1. Comparison of Previous Studies

Author (Year)	Method	Object / Case Study	Testing	Key Findings / Limitation
Zulfikar & Qoiriah (2025)	Agile (Kanban)	Yayasan Umar Faruq Al Arifin	Not stated	Improved transparency and data security; Kanban board clarified workflow, but limited to a single institution.
Utami Dewi & Voutama (2024)	UML-based design	Al-Amanah Mosque financial system	Not stated	Use Case, Activity, Sequence, and Class Diagrams produced a functional, secure system; development method not explicitly Agile.

Djohan <i>et al.</i> (2025)	UML Agile	within	University-level system		Not stated	Mature UML modeling synchronized developer-user understanding; limited to institutional scope.
Andika Isma <i>et al.</i> (2024)	Agile		CV Computer	Speed	Not stated	Enabled real-time, location-independent monitoring; emphasized ease of data input, but single-revenue-stream business.
Fauzan Islamil Haq <i>et al.</i> (2024)	Agile		RT-level community finance		Blackbox	Intensive observation/interviews aligned the system with real needs; small-scale community context.
Ikhsanuddin & Megawaty (2026)	Agile (Scrum)		PT Playground Riatar	(e-ticket booking)	Blackbox	Improved service efficiency and reduced queues on the same object; scope limited to ticketing, not cash flow recording.
Sabila <i>et al.</i> (2021)	Agile (Scrum)		Student organization financial report recording		Not stated	Digitized a manual recording process through iterative sprints; scope limited to a single administrative workflow, no cash-flow monitoring.
Riska Aryanti <i>et al.</i> (2022)	Waterfall		PT Suma Karya Gemilang	(construction)	Not stated	Reduced recording errors and accelerated reporting; rigid sequential method less adaptive to changing needs.
Maulana <i>et al.</i> (2025)	Descriptive-quantitative evaluation		Village Financial System (Siskeudes)		Not stated	Data accuracy and transparency significantly affect system effectiveness; human-resource and infrastructure constraints remain.
Ashifuddin & Neni Lisa Firana (2025)	Waterfall		Kindergarten (early-childhood institution)		Blackbox	Real-time dashboard and automatic reporting functioned as expected; not developed iteratively.
Rahmatika & Edi (2022)	Prototype		Dusun Sidawung treasury		Not stated	Role-based access (admin/resident) improved transparency and accuracy over manual recording.
Anwar <i>et al.</i> (2026)	ISO/IEC 25010 evaluation		PT Teknologi Informatika Solusindo		Quality metrics	Functional suitability 84.29% and usability 85.13%, both "very good"; evaluates quality, not development method.
Nugroho (2022)	SDLC Prototyping	/	Kauman finance	Village	Blackbox	ERD/DFD modeling and role-based access met functional requirements; non-iterative development approach.

As shown in Table 1, Agile-based implementations (Andika Isma *et al.*, 2024; Fauzan Islamil Haq *et al.*, 2024; Ikhsanuddin & Megawaty, 2026; Sabila *et al.*, 2021; Zulfikar & Qoiriah, 2025) consistently achieve operational benefits in real-time cash visibility, administrative transparency, and iterative alignment with user expectations. However, existing applications are largely constrained to organizations with singular or uniform revenue streams. Conversely, while sequential and prototyping approaches (Ashifuddin & Neni Lisa Firana, 2025; Nugroho, 2022; Rahmatika & Edi, 2022; Riska Aryanti *et al.*, 2022) achieve structured record-keeping, their linear lifecycles are less adaptive to mid-development requirements adjustments. Furthermore, while quality evaluation frameworks and UML modeling studies (Anwar *et al.*, 2026; Djohan *et al.*, 2025; Maulana *et al.*, 2025; Utami Dewi & Voutama, 2024) provide rigorous evaluative and architectural baselines, they do not resolve the operational challenges of managing multi-channel daily operational cash flows.

2.3 Research Positioning

Although previous investigations confirm that automated systems mitigate manual accounting errors, existing literature does not address the operational characteristics of children's recreational venues, where daily cash flows derive from heterogeneous operational activities—including entrance ticketing, competitive events, culinary classes, and academic tutoring—rather than a single transactional stream. The most closely related work by Ikhsanuddin and Megawaty (2026) examined PT Playground Riatar using an Agile Scrum lifecycle; however, that study was restricted to customer-facing e-ticket reservations and excluded back-office cash flow reconciliation, category management, and ledger monitoring. The present study addresses this gap by developing a centralized, web-based daily cash flow system that consolidates transactions from all operating

activities of Riatar Playground into a single administrative platform. Methodologically, this study applies Agile development to facilitate iterative feature refinement guided by direct stakeholder feedback, aligning with established empirical practices in applied software engineering. From a modeling perspective, system interactions are structured using UML techniques established in comparable studies (Djohan *et al.*, 2025; Utami Dewi & Voutama, 2024). Functional reliability is verified through black-box testing, consistent with validation standards applied in software engineering literature (Ashifuddin & Neni Lisa Firana, 2025; Fauzan Islamil Haq *et al.*, 2024; Ikhsanuddin & Megawaty, 2026; Nugroho, 2022). By integrating diverse activity-based cash flows into a role-segregated architecture, this research complements the venue's existing ticketing mechanisms and provides an architectural reference for financial system implementation in small service enterprises.

2.4 Review of Technologies, Frameworks, and Algorithms

This study builds upon five foundational concepts in information systems engineering. First, an information system is defined as an integrated set of components that collect, process, store, and disseminate data to facilitate operational control and decision-making (Utami Dewi & Voutama, 2024), providing the structural basis for digitized ledger administration. Second, daily cash flow represents the systematic accounting of immediate cash receipts and disbursements, which is vital for maintaining liquidity and operational solvency in transaction-dense environments (Andika Isma *et al.*, 2024). Third, an operational financial information system routinely records internal organizational transactions to ensure data traceability and auditability (Zulfikar & Qoiriah, 2025). Fourth, the Agile development methodology emphasizes flexibility, collaborative stakeholder engagement, and iterative delivery across short development cycles (sprints), ensuring rapid accommodation of operational requirements (Fauzan Islamil Haq *et al.*, 2024). Fifth, the Unified Modeling Language (UML) serves as a visual modeling standard to represent system interactions and database architectures, bridging conceptual designs with user workflows (Djohan *et al.*, 2025). Complementing these theoretical principles, the selected implementation stack—comprising native PHP, the Bootstrap framework, and a MySQL relational database—has been empirically validated in related web engineering studies within the same institutional context, including an academic archival platform (Handayani & Megawaty, 2024), a construction cost estimation system (Panggabean & Megawaty, 2024), and a public health portal (Salsabilah *et al.*, 2024). The demonstrated reliability and low deployment overhead of this technology stack justify its adoption for Riatar Playground.

3. Methodology

The goal of this applied research is to develop a software artifact—a web-based daily cash flow management system—to assist the operational financial governance of Riatar Playground. Employing a qualitative-descriptive and software engineering methodology, user requirements and operational constraints were examined through direct observation and structured interviews, followed by an iterative development process and systematic black-box testing. The entire lifecycle was executed iteratively in accordance with the Agile development model, ensuring that functional modules continuously adapted to user requirements.

3.1 System Architecture and Design

The daily cash flow system for Riatar Playground is architected to support two distinct user roles, Owner and Admin, each assigned differentiated privilege levels. System interaction initiates through a unified authentication login module. To ensure accessibility and responsive presentation across desktop and mobile devices, the front-end user interface was built using native PHP and the Bootstrap framework. The back-end application logic is executed in native PHP, communicating with a MySQL relational database managed via phpMyAdmin on a local XAMPP environment during the development phase. The operational data flow directs authenticated users from the login module to their respective role-based dashboards, from which authorized modules—transaction recording, category classification, financial reporting, or user management—are accessed, ultimately updating a centralized balance table.

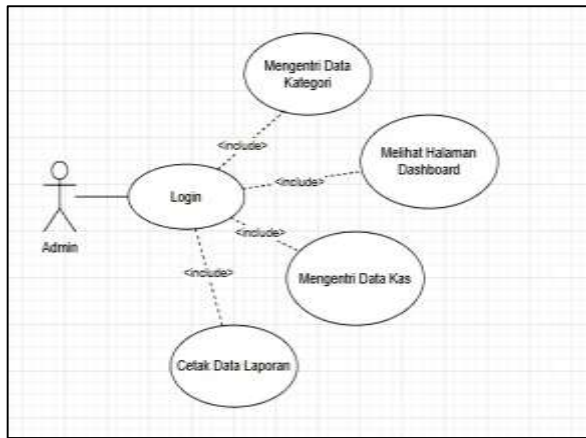


Figure 1. Use Case Diagram for Admin Actor in the Riatar Playground Daily Cash Flow System

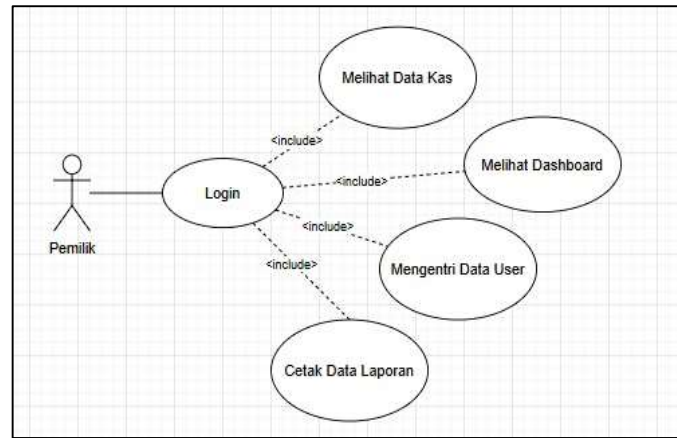


Figure 2. Use Case Diagram for Owner Actor in the Riatar Playground Daily Cash Flow System

The Admin role encompasses four primary functions: 1) viewing the administrative summary dashboard, 2) recording daily cash inflows and outflows, 3) managing master transaction categories, and 4) generating and printing financial reports. Conversely, the Owner role is structured around governance and executive monitoring, comprising four primary functions: 1) viewing the consolidated financial summary dashboard, 2) inspecting the historical transaction ledger, 3) managing user accounts and credentials, and 4) printing periodic financial reports. This role-based separation of duties prevents unauthorized record alterations, thereby safeguarding transaction integrity and operational accountability.

3.2 Development Lifecycle Framework

The methodological framework utilized in this study is the Agile development lifecycle, selected because its iterative structure accommodates evolving user requirements through short feedback loops. The standard Agile lifecycle comprises six phases: Plan, Design, Develop, Test, Deploy, and Review (Fauzan Islamil Haq *et al.*, 2024). In this research, the implementation scope encompassed the first four phases—Plan, Design, Develop, and Testing—while the production deployment and post-implementation review phases were designated for future operational adoption.



Figure 3. Agile Development Lifecycle

3.3 Development Process

Empirical data supporting the development process were gathered through four complementary techniques. Direct observation was conducted at Riatar Playground to analyze the physical cash flow, recording media, and operational bottlenecks experienced by administrative staff. Structured interviews were conducted with three key participants—the owner and two administrative staff members—to identify recording discrepancies, desired platform capabilities, and administrative expectations. A literature review of national and international publications on web-based financial systems and Agile implementations established the theoretical foundation and architectural baseline. Finally, documentation analysis was performed on existing recording media, including manual tally sheets, physical paper receipts, and spreadsheet recaps, to delineate transaction categories and database entities. The development execution was completed over a three-month period across four iterative phases:

- 1) Planning Phase: Executed during the first two weeks of month one, this phase focused on analyzing field observation data and interview transcripts. The core problem was identified as an unintegrated, paper-based recording process prone to ledger discrepancies. User requirements were compiled into a prioritized product backlog based on operational urgency, prioritizing authentication, transaction recording, category management, financial reporting, and executive dashboards.
- 2) Design Phase: Conducted from the second to the fourth week of month one, this phase translated backlog requirements into technical architectural artifacts. These included UML modeling (use case diagrams and activity diagrams), relational database schema specifications across four core tables, and responsive user interface wireframes constructed with the Bootstrap grid system.
- 3) Development Phase: Spanning from the first week of month two to the first week of month three, programming implementation was executed in native PHP connected to a MySQL database. In accordance with the product backlog priorities, development proceeded across five sequential sprints: Sprint 1 implemented the authentication module and role-based access control (Admin and Owner); Sprint 2 implemented daily income and expense entry modules alongside transaction category management; Sprint 3 established the interactive dashboard containing financial metrics and weekly visual trend charts; Sprint 4 implemented periodic financial reporting with custom date filtering and export capabilities; and Sprint 5 delivered the user management module for administrative credential governance. At the conclusion of each sprint, functional increments were demonstrated to the playground owner and administrative staff in a walkthrough session to validate operational alignment and accommodate minor interface refinements prior to the subsequent sprint. A formal, organization-wide user acceptance testing protocol was not conducted within this development phase.
- 4) Testing Phase: Carried out from the second to the fourth week of month three, functional verification was executed using the black-box testing technique. All developed modules were evaluated against predefined valid and invalid test scenarios without inspecting internal source code routines, confirming that each feature executed in conformance with user requirements.

3.4 Dataset Description and Requirements Analysis

Because this software engineering study does not employ an algorithmic machine-learning dataset, empirical field data served as the basis for requirements elicitation. Primary data comprised observational logs and interview transcripts from three participants (the venue owner and two front-desk administrators), while secondary data encompassed manual recording forms, handwritten cash books, and WhatsApp transaction records. Three primary operational bottlenecks were documented: 1) daily entrance transactions were recorded using manual tally marks in notebooks, resulting in frequent recalculation errors; 2) activity registration proofs for competitions and workshops were tracked informally via unsynchronized smartphone screenshots; and 3) periodic financial statements were compiled manually by re-entering paper ledger notes into spreadsheets, introducing reporting delays. Based on these empirical findings, five core functional requirements were established: 1) daily income logging for entrance tickets, private tutoring, and competition fees; 2) daily operational expense logging for venue maintenance and consumable supplies; 3) multi-period financial reporting filterable by day, week, month, or custom date ranges; 4) role-based privilege management dividing administrative recording from executive monitoring; and 5) an executive financial dashboard providing real-time visibility into total revenues, expenditures, and net cash balances.

3.5 Tools and Technologies

The development environment comprised a hardware laptop powered by an AMD Ryzen 5000-series processor, 8 GB RAM, and a 512 GB solid-state drive, accompanied by mobile devices for viewport responsiveness verification. The software development stack is detailed in Table 2.

Table 2. Software Stack

Component	Description
Operating System	Windows 11
Programming Languages	PHP, CSS, and JavaScript
Framework	Native PHP and Bootstrap
Database	MySQL
Web Server	XAMPP (Apache)
Code Editor	Visual Studio Code (VS Code)
Database Management	phpMyAdmin

Following the completion of Sprints 1 through 5, system functionality was evaluated using black-box functional testing. The testing suite evaluated functional outputs against both valid and boundary-invalid input conditions

across seven core operational workflows: authentication, cash inflow entry, cash outflow entry, category classification, periodic report filtering, report export/printing, and session termination.

3.6 Flowcharts, Database Schema, and Interface Design

To formalize system workflows, nine UML activity diagrams were constructed covering all Admin and Owner operations, with each operational flow initiating through credential validation. Figure 4 illustrates the administrative workflow for recording daily cash transactions, demonstrating the login, verification, data entry, and database commit lifecycle characteristic of all system procedures.

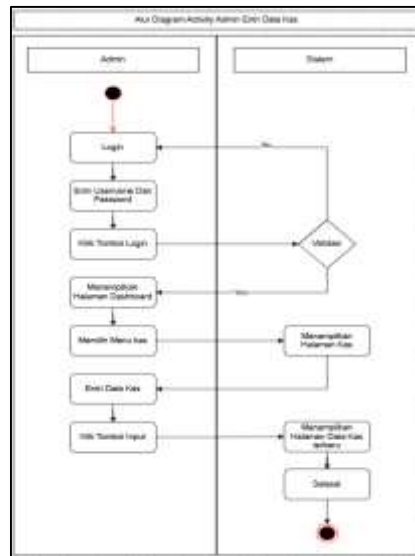


Figure 4. Activity Diagram for Admin Entering Cash Transaction Data

The underlying relational database schema comprises four normalized tables: *tbl_kas* (cash transactions), *tbl_kategori* (transaction classifications), *tbl_user* (user credentials), and *tbl_saldo* (running ledger balance). The core transaction entity, *tbl_kas*, maintains foreign-key relationships with *tbl_kategori* for activity categorization and *tbl_user* for staff attribution, with *tbl_saldo* recalculated upon each committed transaction. Table 3 delineates the data dictionary for the core cash transaction table.

Table 3. Cash Table (*tbl_kas*) Schema

No	Field Name	Type	Length	Description
1	id_kas	Int	11	Primary Key
2	tgl	Date	-	Transaction date
3	keterangan	Int	11	Transaction description
4	pemasukan	Int	11	Cash inflow amount
5	pengeluaran	Int	11	Cash outflow amount
6	saldo	Int	11	Current balance calculation

The front-end user interface was structured to accommodate operational personnel with varying levels of technical proficiency, emphasizing minimal cognitive load, standardized typography, and consistent component placement. The system encompasses nine functional web views, designed using a modular layout that presents operational data logically while restricting unauthorized control elements. During the wireframing phase, high-contrast color coding was applied to differentiate financial inflows from operational expenditures, ensuring immediate data comprehension. In addition, responsive grid breakpoints were configured via Bootstrap CSS to maintain visual hierarchy and component alignment across diverse display viewports, ranging from front-desk desktop monitors to administrative tablet screens. Once a staff member successfully authenticates, the system directs the user straight to an integrated central workspace that consolidates high-frequency administrative controls, eliminating multi-step navigational overhead. Figure 5 depicts the central administrative dashboard, which aggregates key financial metrics, recent transaction entries, and primary navigation links.



Figure 5. Admin Dashboard Page Interface

As illustrated in Figure 5, the interface layout is divided into three functional areas: a persistent sidebar navigation menu on the left, an executive statistical summary card array at the top, and a chronological ledger data table in the main view area. The top cards deliver real-time calculation indicators displaying total accumulated income, total operational expenditures, and current net cash reserves without requiring manual ledger recalculations. Furthermore, the integration of responsive Bootstrap tables enables administrative staff to rapidly filter, sort, and inspect recent ledger entries directly from the primary landing view, thereby mitigating the risk of duplicated or omitted daily cash records.

4. Results and Discussion

4.1 Results

4.1.1 Implementation Results

The web-based daily cash flow management system for Riatar Playground was implemented using native PHP, the Bootstrap framework, and a MySQL relational database. The system is accessible through standard web browsers and accommodates two operational roles, Admin and Owner, each granted access privileges aligned with their daily responsibilities.



Figure 6. Login Page Interface



Figure 7. Admin Dashboard Interface

Users must authenticate through the login interface shown in Figure 6 to access system functionalities. The page provides input fields for registered email credentials and passwords. Upon form submission, the system verifies the credentials against the database. Valid credentials redirect the user directly to the role-specific dashboard, whereas invalid entries trigger a warning alert and prompt the user to re-authenticate. Following successful authentication, administrative personnel are directed to the administrative dashboard shown in Figure 7, which serves as the primary workspace for operational financial tracking. The dashboard presents an executive summary comprising daily revenue, daily expenditures, the current net balance, and the total number of registered users. Visual data components—including a seven-day revenue trend chart, an income-versus-expense proportion graph, a month-to-date financial summary, and a tabular log of the five most recent transactions—provide an immediate overview of venue liquidity.



Figure 8. Owner Dashboard Interface

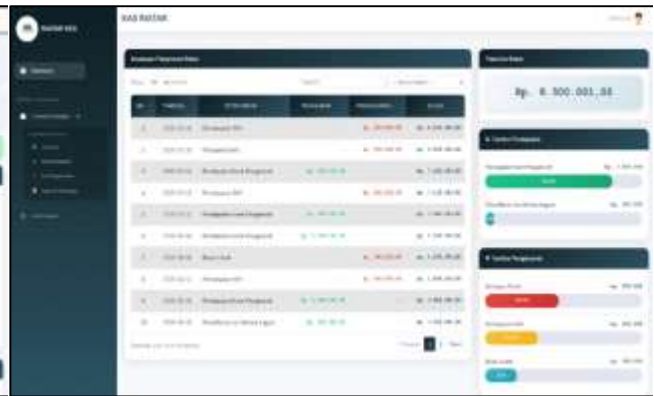


Figure 9. Admin Cash Transaction Entry Page

The Owner dashboard depicted in Figure 8 contains financial summary cards, trend charts, and transaction logs identical to the administrative view. The key operational distinction lies in navigation privileges: the Owner possesses exclusive access to the user management module for system credential administration but is restricted from modifying operational transaction categories. Figure 9 illustrates the consolidated cash transaction page accessible to the administrator. This interface incorporates search filtering, column sorting, and pagination controls to facilitate the recording, updating, and removal of ledger entries. Dedicated sub-modules enable structured categorization of diverse revenue streams—such as entrance tickets, coloring contest registrations, culinary workshops, and private tutoring fees—as well as recurring operating expenditures, including venue maintenance, supplies, and utilities.



Figure 10. Admin Financial Report Generation Page



Figure 11. Owner User Management Interface

To facilitate systematic reporting for venue leadership, the financial report module shown in Figure 10 enables administrative staff to generate periodic statements filtered by specified date ranges. The generated reports summarize total cash receipts, total disbursements, and the closing cash balance for the selected period, complete with integrated print and export capabilities. In alignment with internal control principles separating execution from oversight, the Owner possesses read-only access to transaction ledgers and periodic reports, allowing continuous financial inspection without the ability to alter underlying entries. The user management interface shown in Figure 11 is accessible exclusively to the venue Owner. This module allows the Owner to create, update, and deactivate user accounts, displaying an administrative roster with user names, registered email addresses, and designated role assignments. Restricting account provisioning to the Owner ensures data integrity and prevents unauthorized administrative access.

4.1.2 Testing Results

System functionality was evaluated using the black-box testing methodology to verify that each developed feature satisfied predefined requirements. Functional behavior was assessed by executing seven test scenarios covering valid operational entries and boundary conditions, comparing observed outcomes against expected system behavior. The results are summarized in Table 4.

Table 4. Black-Box Functional Testing Results

No	Test Scenario	Expected Result	Result
1	Login with valid credentials	System displays the dashboard corresponding to the user's role	Valid
2	Login with invalid credentials	System displays a warning message and returns to the login page	Valid

3	Input daily income	Data is saved and displayed on the income ledger page	Valid
4	Input daily expense	Data is saved and displayed on the expense ledger page	Valid
5	Filter report by period	System displays transaction records matching the selected date range	Valid
6	Print/export financial report	System generates formatted report available for printing or export	Valid
7	Logout from the system	User session terminates and system redirects to the login page	Valid

As detailed in Table 4, all seven functional scenarios executed successfully without errors or unexpected exceptions, confirming that the platform meets the functional criteria required for operational adoption at Riatar Playground.

4.1.3 Performance Evaluation and Testing Metrics

Because this study addresses a transactional web-based information system rather than a computationally intensive algorithmic model, performance was evaluated qualitatively against the manual recording procedures it replaces. Unlike the previous paper tally method, the implemented system automatically recalculates running cash balances upon every transaction entry, consolidates disparate revenue streams into a centralized database, and delivers real-time financial metrics through an interactive dashboard.

The primary quantitative metric utilized is the functional test pass rate derived from black-box testing. Across all seven evaluated operational workflows—encompassing user authentication, cash inflow entry, cash outflow logging, periodic filtering, report generation, and session termination—the system achieved a 100% pass rate (Table 4). This confirms that all core functional requirements specified in Section 3 perform consistently under tested operating conditions.

4.2 Discussion

The empirical outcomes demonstrate that the web-based daily cash flow system for Riatar Playground was successfully constructed following the four phases of the Agile development lifecycle: Plan, Design, Develop, and Testing. The platform directly resolves the primary operational bottleneck identified in the Introduction—specifically, the administrative burden and error vulnerability associated with paper tally marks and fragmented WhatsApp payment confirmations. By migrating transaction logging to a relational database, the system mitigates arithmetic calculation discrepancies and standardizes daily cash reconciliation. The interactive dashboard enables executive management to monitor cash balances and revenue trends continuously without awaiting manual end-of-month reconciliations. This finding corroborates the conclusions of Andika Isma *et al.* (2024), who observed that cloud- and web-based financial architectures allow business proprietors to track fiscal conditions remotely and promptly. Furthermore, implementing role-based access control enforces an operational separation of duties between the Admin and Owner. Restricting administrative staff to operational data entry while granting the venue Owner exclusive governance over credential management and strategic oversight enhances data security and internal accountability. This architectural division aligns with the governance recommendations established by Zulfikar and Qoiriah (2025), who highlighted that role differentiation in financial software mitigates internal control risks and record manipulation.

The Agile development framework contributed directly to project success. Conducting five iterative sprints paired with end-of-sprint walkthroughs ensured that functional increments aligned closely with user workflows. This empirical progression supports the observations of Fauzan Islamil Haq *et al.* (2024), who emphasized that iterative Agile cycles facilitate the rapid accommodation of field-level operational needs. Nevertheless, an important boundary condition must be noted regarding the operational validation of this system. Because project execution concluded at the Testing phase, the Deploy and Review stages of the Agile lifecycle were not conducted. Consequently, system evaluation remains bounded to controlled black-box test cases and informal stakeholder walkthroughs, without longitudinal data on everyday user adoption, high-volume concurrent throughput, or transactional robustness under prolonged operational stress. While the 100% test pass rate verifies functional correctness, future research should implement a live pilot deployment to assess sustained user satisfaction and real-world system reliability in daily playground operations. Overall, the system fulfills its functional specifications—spanning revenue and expenditure recording, category management, date-filtered reporting, dashboard analytics, and user administration. The black-box testing results validate that the software artifact is technically stable and prepared to support the financial administration of Riatar Playground.

5. Conclusion and Recommendations

This study successfully designed and developed a web-based daily cash flow management system for Riatar Playground using the Agile development method across four lifecycle phases: Plan, Design, Develop, and Testing. Implemented through five sequential sprints, the platform replaces fragmented manual recording procedures by consolidating daily revenue and expenditure transactions into a centralized relational database. The integrated platform directly resolves the operational challenge of reconciling cash flows across heterogeneous service activities by providing structured transaction categorization, multi-period date-filtered reporting, and interactive dashboard analytics. Role-based access control successfully establishes an operational separation of duties, restricting front-desk administrators to transaction entry while granting the venue Owner exclusive oversight and credential governance, thereby improving data integrity and administrative accountability. Black-box functional testing across seven core operational workflows—including authentication, cash inflow entry, cash outflow entry, category classification, report filtering, report export, and session termination—achieved a 100% pass rate, validating that the software artifact is technically viable and ready for operational adoption. The primary contribution of this research is an integrated financial management architecture that consolidates cash flows from diverse, multi-stream playground activities—encompassing entrance ticketing, play-area rentals, coloring competitions, craft workshops, culinary classes, and private academic tutoring—into a unified digital ledger. This system advances prior work conducted at the same venue by Ikhsanuddin and Megawaty (2026), which was restricted to e-ticket reservations and did not encompass back-office cash flow reconciliation or ledger closing. Methodologically, the study demonstrates that the Agile development lifecycle, when supported by iterative sprint-end stakeholder walkthroughs, effectively accommodates evolving operational requirements in small-scale service businesses, corroborating the findings of Fauzan Islamil Haq *et al.* (2024). Furthermore, the resulting role-based architecture serves as a practical blueprint for small and medium-sized service enterprises seeking to decouple daily recording tasks from managerial financial monitoring within a lightweight system framework.

Several limitations of this study should be acknowledged. First, the development lifecycle was bounded to the Plan, Design, Develop, and Testing phases; production deployment and post-implementation review were not executed, meaning that long-term transactional stability and user adoption under daily operational stress remain unobserved. Second, system verification was confined to functional black-box testing across seven baseline scenarios, leaving non-functional parameters—such as transaction concurrency, cryptographic security, and scalability under high data volumes—untested. Third, requirements elicitation relied on qualitative data obtained from three participants (the owner and two administrative staff members) at a single facility, which may constrain the immediate generalizability of the findings to larger entertainment centers with multi-branch organizational structures. In light of these findings and constraints, several recommendations are proposed for future technical development and academic research: 1) incorporate automated cloud-based database backup and recovery routines to safeguard financial records against server failures or accidental data corruption; 2) develop a dedicated cross-platform mobile application to facilitate location-independent financial monitoring and receipt scanning for venue management; 3) integrate graphical analytical features into the reporting module, such as comparative monthly trend visualizations and expense breakdown charts, to support strategic budgeting; and 4) execute the live deployment and review phases of the Agile lifecycle through an extended operational pilot at Riatar Playground, accompanied by formal user acceptance testing and comprehensive stress testing under concurrent workloads.

References

- Andika Isma, Muhlis, A. K., Ardiansyah, Asriyani, & Fadhilatunisa, D. (2024). Sistem informasi keuangan berbasis web menggunakan pendekatan Agile. *Jurnal MediaTIK*, 6(3), 62–68. <https://doi.org/10.59562/mediatik.v6i3.1474>
- Anwar, C., Farizy, S., & Wijayanto, S. (2026). Implementasi ISO/IEC 25010 dalam evaluasi kualitas fungsional dan usability sistem informasi keuangan: Studi kasus PT Teknologi Informatika Solusindo. *Jurnal Rekayasa Sistem & Pengetahuan Terkini*, 10(2), 3034–3042. <https://doi.org/10.36040/jati.v10i2.17898>
- Ashifuddin, A., & Firana, N. L. (2025). Smart financial system: Rancang bangun sistem informasi keuangan berbasis web pada TK Darma Indra Kaliwungu Selatan Kendal. *Reputasi: Jurnal Rekayasa Perangkat Lunak*, 6(1), 21–26. <https://doi.org/10.31294/reputasi.v6i1.8927>

- Djohan, H., Gusti, I., Pramesti, A., Putri, D., Putu, I., & Suyasa, B. (2025). Analisis dan perancangan sistem keuangan Universitas Primakara menggunakan Unified Modeling Language (UML) dengan metode Agile. *Jurnal Sistem Informasi dan Ilmu Komputer Prima (JUSIKOMP)*, 8(2), 280–287. <https://doi.org/10.34012/jurnalsisteminformasidanilmukomputer.v8i2.6415>
- Fauzan Islamil Haq, R., Pandiya, R., & Setyadi, R. (2024). Rancang bangun sistem informasi keuangan tingkat RT menggunakan metode Agile. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(1), 48–56. <https://doi.org/10.36040/jati.v8i1.8558>
- Handayani, F. K., & Megawaty. (2024). E-archive academic data at SMK Negeri 2 Sekayu. *International Journal Software Engineering and Computer Science (IJSECS)*, 4(1), 1–12. <https://doi.org/10.35870/ijsecs.v4i1.1583>
- Ikhsanuddin, A. N., & Megawaty. (2026). Rancang bangun aplikasi booking online e-tiket berbasis web di PT Playground Riatar. *Jurnal Riset Sistem Informasi dan Teknologi Informasi (JURSISTEKNI)*, 8(1), 1025–1037. <https://doi.org/10.52005/038rd772>
- Maulana, R., Rahmi, Z., Rasyada, R. D., & Najmi, M. (2025). Analisis efektivitas penggunaan sistem informasi keuangan di pemerintahan desa. *Jurnal Teknologi dan Informasi*, 4(2). <https://doi.org/10.59086/jti.v3i3.1044>
- Nugroho, A. (2022). Analisis dan perancangan sistem informasi keuangan pada Desa Kauman berbasis website. *Jurnal Ilmu Siber*, 4(1), 1–10. <https://doi.org/10.24076/joism.2022v4i1.769>
- Panggabean, W. P., & Megawaty. (2024). Implementation of design thinking UI/UX construction RAB application at PT PLN (Persero) UID S2JB Lahat. *International Journal Software Engineering and Computer Science (IJSECS)*, 4(3), 1173–1183. <https://doi.org/10.35870/ijsecs.v4i3.3120>
- Rahmatika, D. P., & Edi, S. W. M. (2022). Pengembangan sistem informasi keuangan berbasis web untuk bendahara Dusun Sidawung. *Indonesian Journal on Networking and Security (IJNS)*, 1(1), 33–48. <https://doi.org/10.24246/itexplore.v1i1.2022.pp33-48>
- Riska Aryanti, Puspitasari, D., Efendi, R. R., & Esmeralda, R. N. (2022). Implementasi model Waterfall pada pembangunan sistem informasi keuangan proyek. *Jurnal Teknik Komputer AMIK BSI*, 8(2), 174–180. <https://doi.org/10.31294/jtk.v4i2>
- Sabila, H., Praptono, B., & Arini, I. Y. (2021). Perancangan aplikasi pencatatan laporan keuangan dengan menggunakan metode Agile Development Scrum. *Journal of Information Systems and Informatics Engineering (JOISIE)*, 5(2), 67–74.
- Salsabilah, E., Megawaty, Fatoni, & Amalia, R. (2024). Health information media for outpatient services at Puskesmas OPI Palembang based on a website. *International Journal Software Engineering and Computer Science (IJSECS)*, 4(3), 1137–1149. <https://doi.org/10.35870/ijsecs.v4i3.3132>
- Utami Dewi, A., & Voutama, A. (2024). Implementasi UML dalam perancangan sistem informasi keuangan Masjid Al-Amanah berbasis web. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(5), 10300–10308. <https://doi.org/10.36040/jati.v8i5.11026>
- Zulfikar, M., & Qoiriah, A. (2025). Perancangan dan pembangunan sistem informasi keuangan Yayasan Umar Faruq Al Arifin berbasis website menggunakan metode Agile pendekatan Kanban. *Journal of Informatics and Computer Science (JINACS)*, 6(4), 1120–1130. <https://doi.org/10.26740/jinacs.v6n04.p1120-1130>