privietlab.
RESEARCH & PUBLISHING

Priviet Social Sciences Journal

Volume 6
Issue 1 January, 2026

Article 62

29-01-2026

Analysis and design of a web-based integrated inventory information system using the PIECES framework: A case study of PT. Asia Persada Nusantara

Uya Asy Syuura Anandri & Ilyas

To cite this article: Anandri, U. A. S., & Ilyas, I. (2026). Analysis and design of a web-based integrated inventory information system using the PIECES framework: A case study of PT. Asia Persada Nusantara. *Privlet Social Sciences Journal, 6*(1), 702-724.
https://doi.org/10.55942/pssj.v6i1.1514

To link to this article: https://doi.org/10.55942/pssj.v6i1.1514

Private Social Sciences Journal
E-ISSN 2798-866X
P-ISSN 2798-6314
https://doi.org/10.55942/pssj.v6i1.1514

Volume 6
Issue 1
January Edition 2026

Analysis and design of a web-based integrated inventory information system using the PIECES framework: A case study of PT. Asia Persada Nusantara

Uya Asy Syuura Anandri* **& Ilyas**

Information System, Faculty Engineering and Computer Science, Islamic University of Indragiri
Jl. Provinsi Parit 1, Tembilahan Hulu, Indragiri Hilir, Riau, Indonesia
*e-mail: uyaanandri@unisi.ac.id

Received 09 December 2025
Revised 26 January 2026
Accepted 29 January 2026

ABSTRACT

This study aims to analyze and develop a systematic or analytical web-based integrated inventory information system for PT. Asia Persada Nusantara, analytical used by PIECES framework. Based on a case-study approach, both with the SDLC methodology with a prototyping model, the requisite data were acquired by a complete review of documents and by conducting interviews with stakeholders. The Object Oriented Tool Development was modeled using the Unified Modeling Language (UML) to define the functional and non-functional requirements. The resultant design captures role-based authentication, stock reservation systems, processing of goods receipts including the processing of delivery notes, keeping of discrepancies, payment and reconciliation systems, and role-based audit trails as a way of enforcing operational control. The shared schema of the relational database using ERD supports the atomic transactions of the key turbines (reservation, transaction creation, commitment) and helps to reduce the risk of overselling and information inconsistency. It is proposed that a staged implementation plan (Minimum Viable Product) should be used with an initial focus on real-time inventory and reservation logic to achieve more immediate operations payoffs. The indicators of future improvement in stock accuracy and process efficiency are stated by impact assessment, whereas the real benefits depend on piloting testing, initial data quality, and organizational readiness. This research provides a technical map that is suitable for developing prototypes and validating them empirically.

Keywords: Inventory Information System; PIECES; Stock Reservation; UML; ERP; Audit Trail; SDLC Prototype Model

1. INTRODUCTION

The fast advancement in information technology has radically changed the working environment in the trading business but has also revealed sustained shortfalls in the procedural areas more so in distribution and warehousing where operations are geographically divided (Lopulalan et al., 2025). The internal documents and audit of operations of PT Asia Persada, a general trading organization that serves various sub-districts and operates within the area of Indragiri Hilir, reveal a trend of tangible and consistent problems: the existence of repeated discrepancies between physical and recorded data, the variability of warehouse coordination, the discrepancy between the amount invoiced and the receipt, and lack of traceability of its operations. These inadequacies equate to staff day-in-day-out operational overhead as manual reconciling and result in a worsening customer experience as sales are delayed, increasing volumes of claims, and a financial constraint in the form of cash-flow drag and the risk of sales lost to overselling. These tangible effects indicate that the technological solutions are insufficient, but solutions should be customized to the local connectivity limits, human input rhythm, and managerial behaviors which are inherent in the multi-warehouse and resource-constrained environments (Anandri, 2025).

Relevant foundations come with the academia and practice-based guidelines of inventory-control models, ERP design principles, software engineering models like UML modelling but they face two noticeable constraints in the current context (Ilyas et al., 2024). To start with, the assumption of centralised system, consistent, high-quality data and constant connectivity is based in numerous of the studies; this weaknesses the application of the models to the small-to-medium and regionally dispersed traders who conduct business on intermittent connectivity and intensive manual operations (Anandri & Arkan, 2025). Second, although there is an apparent relative lack of empirical work, which would combine a PIECES-informed diagnosis with concrete design artifacts (UML diagrams, entity to reserve logics, and reservation logic, etc.), and an otherwise viable implementation plan (minimum viable product and pilot key performance indicators) in such a setting, business and technical requirements frameworks like PIECES provide a valuable map to follow. Concisely, the literature has not fully addressed the translation of the PIECES-oriented designs to functionable prototypes and quantifiable improvements upon implementation even in pilot trading settings, namely, multi-warehouse trading (Anandri, Arkan, et al., 2025).

This study aims to fill that gap by sitting at the intersection of diagnosis, design, and implementation. The study will be structured in such a way that it has three objectives that are connected to one another. The first step in defining the root causes of inventory variances, reconciliation failures, and process bottlenecks of PT Asia Persada, on the basis of the PIECES framework (Performance, Information, Economy, Control, Efficiency, Service) based on company documents, participant observations, and semi-structured interviews by referring to company documents, participant observation, and interview. Second, the design of a web-based (stated in artifacts of UML) Integrated Inventory Information System (IIS) (implemented via a relational database schema) containing role-based authentication, reservation/commit logic to fulfil the order structure, goods-receipt validation, and audit trail of a transaction. Third, to present a staged MVP implementation plan that has specific pilot KPIs (e.g., inventory variance rate, the time-to-reconcile, order-to-ship latency, and customer-claim frequency) and actionable recommendations that could be implemented and monitored by managers on a case of a single warehouse pilot and then scaled (Anandri, Arkan, et al., 2025).

Methodologically, the research design is a one-case explanatory design that combines a documentary study, minimal participant observation, and semi-structured interviews with strategic operational stakeholders. Descriptive statistics would help a transaction log and operations report summarize stock discrepancies and processing delays, and qualitative data are thematically coded to reveal user requirements and process failure modes. This information can be converted into relevant and implementable technical specifications using UML and ERD artifacts. The system design follows the software development life-cycle prototype (MVP) model so that any proposed modules can be piloted, iterated, and tested based on the defined KPIs. These convergence methods were intentionally selected to relate certain confirmed design choices, such as reservation expiry policies, audit-loging details, and input limitations on the user interface, to measurable operational achievements.

This study makes three contributions. Pragmatically, it provides a deployable IIS architecture that suits a multi-warehouse trading company with declarative specifications on the modules and an MVP roadmap that managers can implement. It conceptually and methodologically enhances the literature around the PIECES-based approach, showing how diagnostic categories can be articulated as design artifacts and pilot metrics in a resource-limited environment. Finally, this study provides specific and prioritized recommendations (who is to bear responsibility, implementation sequencing, and monitoring mechanisms) to close the gap between technical design and rational operational practice. Accomplishing these goals will enable managers to minimize the amount of errors and rework, increase customer service performance, and strengthen financial controls, which are smart outcomes of undisputed importance in the case of PT Asia Persada Nusantara and similar regional trading firms.

2. LITERATURE REVIEW

2.1 Inventory information systems in distributed and multi-warehouse environments

Placing inventory information systems are important systems that form the basis of material flow coordination, meticulous transaction recording, and the development of operational decisions in trade and distribution businesses. The research evidence indicates a reliable pattern of inaccuracy in inventory data as the cause factor leading to the formation of stockouts, surplus inventories, delayed shipments, and worsening of the customer satisfaction (Alfassa et al., 2023; Gunasekaran et al., 2017). These issues become even more complex in multidimensional warehouse settings, where the geographical locations of the operational units are widely spread, and the effectiveness of data synchronization strongly depends on a well-integrated system and trained operations.

The existing studies on inventory systems have been largely focused on centralized enterprise applications of the ERP systems on the basis of standard operations and strict IT governance (Hendricks et al., 2007; Rianto et al., 2023). Despite their effectiveness in large companies, such assumptions are often broken in practice when implemented in small- and medium-sized trading companies that have several warehouses with dissimilar working processes and limited technological resources. The empirical studies have shown that decentralized warehouses tend to rely on either manual records or half-baked tools, and thus, there is a high likelihood of inconsistency in data and delayed reconciliation schedules (Kembro et al., 2017). As a result, there is a need for inventory information systems that are clearly designed to suit distributed operational circumstances and are not based on the idyllic paradigms of centralization.

2.2 Real-time inventory control, stock reservation, and reconciliation mechanisms

There is no doubt that today, real-time inventory management has become an essential part of supply chain responsiveness and service-based reliability in the modern aspects of supply chain operations. The key advantage of perpetual inventory given that the organization keeps the stock level updated according to the incoming and outgoing transactions compliance to such system would help significantly lower the information latency and phantom inventory (Atieh et al., 2016). Empirical research has shown that real-time visibility has a meaningful effect on improving accuracy in order fulfillment and reducing operational uncertainty, especially in business environments with high transaction volumes.

Stock reservation mechanisms are more of a complementary insurance against overselling and transaction disputes since they mark the inventory units temporarily held on behalf of the confirmed orders, thereby averting these risks. Studies on order -management suggest that reservation-based logic is particularly effective in conditions which have multiple users and multiple location where simultaneous access to inventory data is a matter of course (Huang et al., 2019). When such mechanisms are not employed, the organizations suffer increased order cancellations as well as concomitant loss in customer satisfaction.

The payment reconciliation and invoice matching add additional supporting support to the integrity of the transactions, and the financial transparency. It has been demonstrated that automated reconciliation processes can reduce the amount of manual work, reduce errors, and improve visibility of cash flows (Ouyang & Jiao, 2021). The combined design of real-time inventory notifications, booking algorithm and reconciliation systems, therefore, creates an integrated control framework, which facilitates operational effectiveness and financial competitiveness at the same time.

2.3 Audit trails, control, and accountability in information systems

Audit trails serve as a pillar of the internal control systems in any information system, as they are a fundamental log of transactional provenance, which includes the identity of the actor, the time, and location, and the compromised information, which mitigates the accountability and parallel traceability. As per the scholarly studies, there is a consistent finding that sound audit trails enhance the integrity of the data, facilitate compliance with regulations, and institutional trust in the outputs of the system (Alles et al., 2018).

Auditability-wise, its importance increases at every possible point at which the interfering or overruling activities by a person as a part of a manual control are guaranteed. The empirical evidence indicates that the information systems that lack systematic audit trails are disproportionately vulnerable to fraud, information manipulation, and other unintentionally misleading forms of information (Vasarhelyi et al., 2015). This is the result of integrating audit trail capabilities with transactional processes, as opposed to segregating them to secondary logging, and thus the strength of controls, as well as compliance by people, is enhanced. This position is consistent with current scholarly literature that states that mechanisms of control should be viewed as fundamental system architecture elements, as opposed to an a posteriori additions.

2.4 Object-oriented analysis-design and UML-based system modeling

The object-oriented analysis and design can provide a strict, systematic way of bridging the requirements of an organisation, with the refined needs, into structures of systems that are maintainable and modular. Object orientation foregrounds (object orientation) anticipates such principles as encapsulation, abstraction, and reuse, which are especially relevant in terms of complex systems defined by a variety of actors and intricate processes (Booch et al., 2007).

Unified Modeling Language (UML) has over time become known as de facto standard notation in terms of representation of system requirements and system designs. Use-case diagrams are used in the elicitation of the requirements which explain the relationships that follow in connection to users and the system. Class diagrams, in their turn, contain the entities of domain, and the relationships between them, and the sequence and activity diagrams make the processes dynamics, and flow of control. The empirical studies have proved that the UML based modelling facilitates communication between stake holders and helps in reducing uncertainty at the developmental stage, specifically in the initial stages of the design (Dobing & Parsons, 2006).

However, much of the known literature supporting UML has focused on technical fullness as opposed to situational pertinence. The literature on prioritisation of UML artefacts on piloting implementations on resource constrained organisations is scanty. This gap highlights the urgent need to match UML modelling practices to real world development strategies - e.g. iterative prototyping and staged implementation in an attempt to achieve real organisational value.

2.5 SDLC and prototype-based development approaches

The Systems Development Life Cycle (SDLC) provides a formal, strict guide to the process of planning, development and implementation of information systems. Traditional SDLC paradigms, which should be described as linear and sequential, are productive in the conditions when the requirements of a project are well-established and stable, however, critics often note that they are rather rigid in dynamic and changing environments (Anandri et al., 2025; Sommerville, 2016).

The model of SDLC called prototyping was developed based on the idea of overcoming this weakness and providing early user involvement and continuing improvements. Empirical work has been consistent to show that prototyping leads to improved accuracy of the requirements, enhanced user acceptance, and reduced risk of wrong development particularly those systems where human interaction and procedural complexity are high (Rai et al., 2002). When the developers allow users to play with developing versions of the system, this will help them detect the areas of usability and understand the areas of wrong assumptions at the earliest stage of the development cycle.

However, the empirical data about the implementation of SDLC-prototype approach in the sphere of small and medium trading companies is minimal. The empirical literature has been unequal in terms of focusing more on commercial software products as opposed to the information systems embedded in the organization that supports day to day operational processes. The discovery of this evidentiary gap is why case-based research is highly needed to demonstrate how prototype-based SDLC-based methodologies can be practically utilized given practical constraints of an organization.

2.6 The PIECES framework and research gap positioning

PIECES model, which includes Performance, Information, Economy, Control, Efficiency, and Service, is a highly proven tool that allows diagnosing the organisational and information-system problems. It promotes systematic grouping of the issues and assists in the structured requirement analysis as the operational symptoms are correlated with the deficits of the system (Whitten & Bentley, 2007).

However, PIECES has found use as a descriptive research diagnostic instrument throughout the first stages of analysis used in the past. Less studies have quantified its use into the design development and evaluation stages and mapped the dimensions of PIECES explicitly on design artefacts and measureable results. As a result, there is a gap in methodology in terms of how PIECES can be operationalised to take place beyond problem identification.

The literature review comes out with three major gaps. First, there is a contextual void concerning multi-warehouse trading firms through the prism of the decentralised and resource-limiting situations. Second, an epistemic discrepancy remains in the incorporation of PIECES analysis of the object-oriented design artefact and prototype-based assessment. Third, there is a perceived gap in practicability in that there is no empirical evidence that shows how these integrated approaches must serve the staged implementation and pilot deployment. Such gaps are addressed in this study which consists of integrating PIECES analysis, UML-based and SDLC-prototype methodology in a real organisational environment.

3. MATERIALS AND METHOD

3.1 Methodology

The current study follows a case study, explanatory research approach, which focuses on PT Asia Persada nusantara. The given case study could be regarded as an instrument to illustrate the implementation of modern software development practices in a manufacturing environment. The methodology used in the development is grounded in the Systems Development Life Cycle (SDLC) and a Prototype Model of the SDLC in particular. In this study, the process is divided into the following stages, which include: requirements elicitation, prototype development, user evaluation, prototype refinement and elaboration of implementation specifications. All the stages are defined in such a way that there is a sequential flow between the conceptualization and the final artifacts.

In the requirements elicitation stage, the PIECES framework is operationalised hence ensuring that the requirements of performance, information, economy, control, efficiency and service requirements are both identified and ranked. This systematic methodology lifts the quality of the requirement analysis and eases the process of conforming to the stakeholder anticipations.

The reason behind the adoption of the prototype model is that it will provide the end users with a preview of the potential system at an early stage and in a tangible form. This type of artefact will help in early functionality testing, elicit user feedback in an iterative process, and finally remove chances of expensive design changes later into the project life-cycle. The analytical and design activities are built on an object oriented paradigm underlining the construction of UML artefacts. Such artefacts are platform-neutral hence facilitating easy translation to implementation code and database schema designs. This theoretical base levels up the maintainability and extent of the end product of the software. Figure 1 below exemplifies in a graphic manner the entire SDLC- prototype workflow applied in this venture hence illustrating the interrelations of the different stages of the development process.

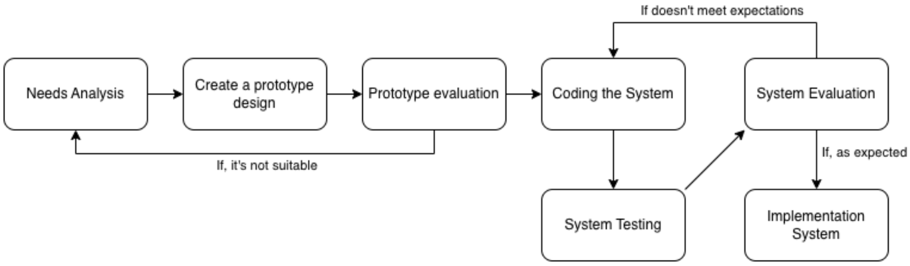

Figure 1. Software Development Life Cycle (Prototype Model)

As shown in Figure 1, the prototype development method translates an early functional model into the final software through a short, iterative cycle that emphasizes rapid delivery and continuous user feedback. The process begins with needs analysis, where stakeholder's requirements and critical business rules are identified; these findings are immediately used to create a prototype design that captures the core interactions and user interface elements. When the prototype is accepted, coding of the actual system proceeds with the prototype serving as a practical specification for developers, which typically shortens development time and lowers cost by clarifying requirements early.

3.2 Data analysis

In the data analysis, a triangulation methodology is applied to protect the reliability and validity. The sources of data include internal records and logs, participant observation records, and semi-structured interview transcript. The strategy of analysis is dichotomous: a) Qualitative analysis: Coding: Interpretation of the interview transcripts and the observations is done by a thematic analytic approach. Based on the PIECES framework, coding then follows on an open coding, then comes axial coding, followed by selective coding. To improve the accuracy of coding, codebooks are built and inter-rater reliability is determined on a representative sample of transcripts. The emergent themes are used in outlining process failure modes, user requirements as well as acceptance constraints (Anandri & Prasetyo, 2025); b) Quantitative analysis: Transactional records are extensively processed and cleansed to give descriptive indicators that include frequency of inventory discrepancies, average order reconciliation time, order-to-ship latency and the share of orders that are victims of reservation conflicts. Multiple descriptive statistics such as counts, percentages, means and medians are provided. In cases where the data availability allows it, comparative pre- post analyses that are conducted with either paired-samp tests or non-parametric alternatives are recommended in pilot evaluation cases (Anandri, Nabil Arkan, et al., 2025).

The triangulation of data is clearly included in the results: no less than two independent data sources support every major empirical statement (e.g. one document and an interview, an observation and a transactional log). All the data-processing steps, transformations and calculations of key-performance indicators are captured on an appendix to ensure reproducibility.

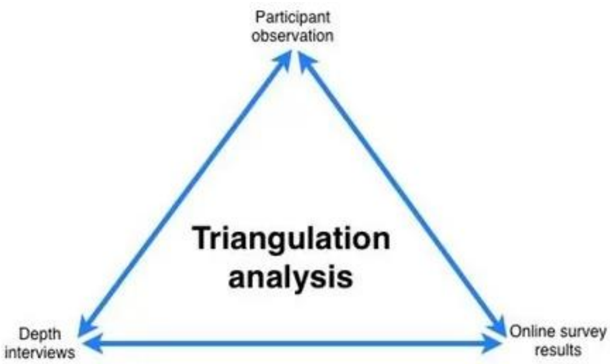

Figure 2. Triangulation Analysis

Figure 2 shows how the validity of the findings is obtained by means of triangulation of the data sources. The outcomes obtained after the document review have been analyzed compared to the results of the interview and observation and then cross-examined by the professionals, academic experts or information systems practitioners who are familiar with logistics and accounting to evaluate the viability and applicability of the design recommendations. Research ethics: a fundamental issue was the protection of sensitive information where any data were anonymised and no authorisation should be provided other than academic and system-design needs as determined by the data owner whilst the technical recommendations were developed to ensure data security, access control and audit trail needs. The deliverable of the research such as the recorded PIECES analysis, functional specifications documentation, UML diagrams, database models and initial interface prototyping had to be arranged in such a way that it could be directly used as pilot-testing material in one of the selected warehouse before full deployment of the system.

3.3 Tools Development

This study applied an object-oriented paradigm in the process of design development where the Object-Oriented Analysis (OOA) was utilized in the analysis phase and Object-Oriented Design (OOD) in the design phase. The reason why the object-oriented approach was chosen was that it has the ability to model real-life business entities i.e. objects and their attributes and operations in such a way as to simplify the process of translating the functional and non-functional requirements into a modular and maintainable software design (Anandri, Arkan, et al., 2025).

The modeling language of choice was taken to be the Unified Modeling Language (UML). UML, which consists of a large series of diagrammatic conventions that are well in use throughout the software industry, allows visualization, design, and documentation of complex systems. The outstanding strength that it has is its ability to generate platform-neutral models and therefore, UML artifacts can offer clear implementation instructions that cuts across individual hardware platforms, operating systems, or programming languages (Anandri, Arkan, et al., 2025; Prasetyo et al., 2025).

3.3.1 Object Oriented Analysis (OOA)

The OOA element was implemented with the detection of the actors and the relevant usage scenarios, then the initial domain classes and their relationship were extracted through Use Case diagrams and Case Diagram. The OOA deliverables that were obtained included a catalogue of domain classes, detailed attributes specifications and interaction scenarios therefore providing a foundation on which further design activities commence.

3.3.2 Object Oriented Design (OOD)

The OOD phase built on these bases and defined system structure and behaviour in terms of a set of UML diagrams: (1) Class Diagram specifying class structures, attributes, operations, and inter-class relationships (inheritance, association and aggregation); (2) Sequence Diagram specifying message flow and temporal sequence of events among objects in critical situations; (3) Activity Diagram specifying business process flow or complex system tasks; and (4) State Diagram specifying state transitions of objects with stateful behaviour.

All diagrams were rigorously tended to, as per the traditional UML notation, given a formal name, a version tag so as to enable its identification, and used both as a prototype writing guide and as a testing tool to verify the findings of interviews, observations and analysis - a task akin to triangulation of data. Through this, the modeling documents serve as a requisite channel connecting the expectation of the stakeholders to the code elicited in the implementation. To promote the development of productivity and team rigor the UML artifacts were archived and organized in the order to make use of version control, commenting of changes, and full-inclusive attachments to facilitate auditability and reproducibility of research results. At any reasonable point, the diagrams can also be exported as the appendix to the research report, which would improve the persuasiveness of the design choices and enable an in-depth technical examination.

3.4 PIECES Framework

The PIECES framework (Performance, Information, Economy, Control, Efficiency, Service) was used as a class-ifying tool in conducting analytical work on the system requirements. Operationalisation of functional requirements was then performed through UML and design artefacts have been generated which are use-case diagrams, class diagrams, sequence diagrams, activity diagrams and state diagrams, which illustrate the interactions between actors and the system required data structures (Anandri, Arkan, et al., 2025). The relational design and normalisation steps were used in the creation of the database schema to guarantee the integrity and consistency of the data stored on the database (Arkan & Anandri, 2025). The study was based as an analytical scaffold on a Problem Analysis Model that is a synthesis of problem identification and evaluation steps including a causal mapping of decision nodes and key personnel and all of which is discussed in the context of the PIECES framework.

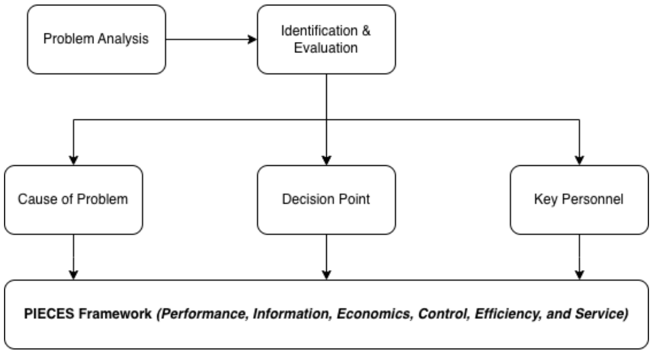

Figure 3. Problem Analysis Model Using PIECES Framework

As shown in Figure 3, The Problem Analysis Model with the PIECES Framework has been tabulated. This diagram explains the logical flow of the process that will start with problem analysis (data collection and reduction) and then move to identification and review of operational problems that will be mapped to three main outputs: the root cause of the problem (e.g., manual input or lack of inter-warehouse communication), decision points (where the critical moments when the decision has to be made occur e.g. when the delivery note is verified or the quantity is verified before shipment), and key personnel (the organisational actors e.g. warehouse staff, sales personnel and finance).

The bottom of the model presents the dimensions of PIECES as a form of evaluation matrix with each issue being connected with the respective technical and business requirements. An example is that a stock discrepancy is plotted to Information and Control, but the time of order processing is plotted to Performance and Efficiency. The model has two roles: it can be used as a framework to operationalize the research instrument (write interview questions and observational protocols) and as a guideline to transform report findings into quantifiable system requirement specification.

3.5 Implementation planning and pilot evaluation

This study applied an object-oriented paradigm in the process of design development where the Object-Oriented Analysis (OOA) was utilized in the analysis phase and Object-Oriented Design (OOD) in the design phase. The reason why the object-oriented approach was chosen was that it has the ability to model real-life business entities i.e. objects and their attributes and operations in such a way as to simplify the process of translating the functional and non-functional requirements into a modular and maintainable software design (Rianto et al., 2023).

The modeling language of choice was taken to be the Unified Modeling Language (UML). UML, which consists of a large series of diagrammatic conventions that are well in use throughout the software industry, allows visualization, design, and documentation of complex systems (Anandri, Arkan, et al., 2025). The outstanding strength that it has is its ability to generate platform-neutral models and therefore, UML artifacts can offer clear implementation instructions that cuts across individual hardware platforms, operating systems, or programming languages (Aswintama et al., 2025).

The OOA element was implemented with the detection of the actors and the relevant usage scenarios, then the initial domain classes and their relationship were extracted through Use Case diagrams. The OOA deliverables that were obtained included a catalogue of domain classes, detailed attributes specifications and interaction scenarios therefore providing a foundation on which further design activities commence. OOD phase built on these bases and defined system structure and behaviour in terms of a set of UML diagrams: (1) Class Diagram specifying class structures, attributes, operations, and inter-class relationships (inheritance, association and aggregation); (2) Sequence Diagram specifying message flow and temporal sequence of events among objects in critical situations; (3) Activity Diagram specifying business process flow or complex system tasks; and (4) State Diagram specifying state transitions of objects with stateful behaviour.

All diagrams were rigorously tended to, as per the traditional UML notation, given a formal name, a version tag so as to enable its identification, and used both as a prototype writing guide and as a testing tool to verify the findings of interviews, observations and analysis - a task akin to triangulation of data (Ramadhan & Mansur, 2024). Through this, the modeling documents serve as a requisite channel connecting the expectation of the stakeholders to the code elicited in the implementation.

To promote the development of productivity and team rigor the UML artifacts were archived and organized in the order to make use of version control, commenting of changes, and full-inclusive attachments to facilitate auditability and reproducibility of research results. At any reasonable point, the diagrams can also be exported as the appendix to the research report, which would improve the persuasiveness of the design choices and enable an in-depth technical examination.

4. RESULT

The findings of the current research are explained based on the analytical and design stages as Methodology in Chapter 3. The reduction and classification of case study information, problem mapping using the PIECES Framework, the definition of functional and non-functional requirements, and system illustrations such as UML models, database layouts, and user interfaces. The main body of evidence found in the serious case study of PT Asia Persada, noted the company's operational problems, namely the inability to report correctly, the lack of stock synchronization in the warehouse, and discrepancies between customer invoices and payment records. The findings and resulting design reflect the final product of the identified condition mapping.

4.1 Analysis PIECES Framework Result

According to the extraction of data collected based on the case study materials and its consequent verification using analytical instrument, as per the Problem Analysis Model as observed in Figure 3, the major findings were then systematized onto the PIECES plot, as shown in Table 1 below.

Table 1. Analysis PIECES Framework Result

Component	Findings / Needs
Performance	The system must be responsive, realtime stock updates between warehouses, authentication of each employee before activities.
Information	Transaction report, delivery history, payment/receivable reconciliation, employee activity log, stock difference notification.
Economy	Minimise the cost of manual error (re-entry), reconciliation automation to reduce accounting time.
Control	Per-employee login, role-based access (admin, accounting, warehouse, sales), audit trail, stock validation before delivery.
Efficiency	Automation of proof making, structured transaction input (form), batch stock update when receiving/sending.
Service	Customer portal to check order/receipt status, periodic reports to the central management.

Source: Processed from primary data (2026)

4.2 Implications Object Oriented Analysis (OOA)

The basic assumption of OOA is based on the concept of objects: discrete entities that unite both data structures and related behaviours in a most coherent structure. Through defining these objects, the analyst will be able to represent in a way that fits semantics of the domain both the static and dynamic components of the system. In reality, OOA concept is an analytical prism that examines the criteria that are present on the system, whilst being faithful to the classes and objects that are met during the modelling process.

4.2.1 Use Case Diagram

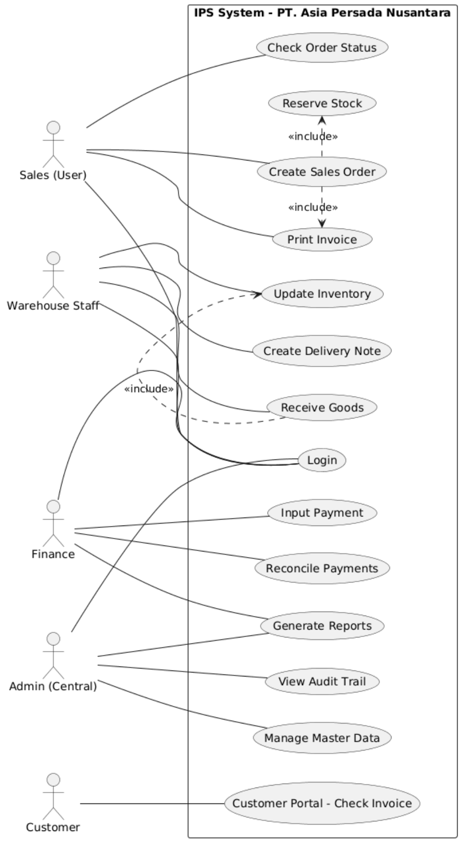

Figure 4. Use Case Diagram

Source: Processed from study case

Figure 4 below indicates the overall Use Case diagram of the PT. Asia Persada Nusantara IPS system that identifies the boundaries of the system and lists the key actors (Sales, Warehouse Staff, Finance, Admin (Central), and Customer) with the primary services each actor may offer that are further supported with the primary services of each actor. The diagram explains the interrelationship between core functions such as authentication, automatic execution of the stock reservation process and invoice printing after the creation of a sales-order, the goods-receipt process which automatically updates the inventory, the payment recording and reconciliation process, and the master-data management and integrated reporting process within the functional system flow. The corresponding relationship (Figure 4) signifies that the order-creation process is never a standalone event that does not involve additional technical activities like stock checks, reservation, and invoice preparation; simultaneously, the goods-recept flow would require verification of the delivery note and update of inventory. This general framework highlights the division of duties among the actors and acts as a backbone to the role-based access control schemes and audit trails, which are non-functional requirements that need to be met throughout the planning of the system.

4.2.2 Case Diagram Sales

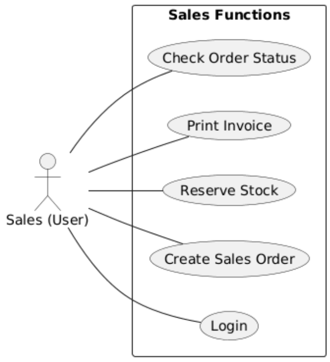

Figure 5. Case Diagram Sales

Source: Processed from study case

Figure 5 shows a detailed case diagram of Sales actor and outlines the Sales role in the entire sales lifecycle which begins with the authentication and ends with the status monitoring of orders. In the functional narrative, the Sales representative will need to authenticate to receive the necessary authorisation to enter customer order information by selecting a suitable warehouse and product; the system will automatically authenticate the data, and the stock reservation process will begin automatically therefore making sure that inventory is available. This order recording sequence is then completed and this gives rise to the creation of a transaction record and an invoice is produced and can be printed or downloaded accordingly. The system also provides a feedback channel, which is communicated by availability messages or a deniability in case of inadequate stock and thus the Sales representative can make the required corrective changes on the order or suggest a back-order. During the post-creation stage, the Sales and the Customer still have that ability to check on the order status and this status shows various progressive stages like reserved, packed, shipped and delivered thus enabling the Sales representative to coordinate follow-up processes.

4.2.3 Case Diagram Warehouse Staff

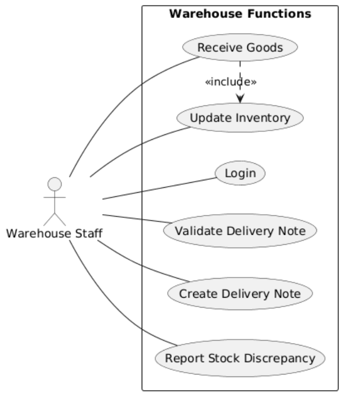

Figure 6. Case Diagram Warehouse Staff

Source: Processed from study case

As shown at Figure 6 outlines a case diagram that summarizes the duties of the role of the Warehouse Staff role, consequently predetermining a set of functions of the operational nature: acceptance of deliveries, confirmation of delivery notes, adjustment of inventory, a new delivery note, and a systematic reporting of stock differences. This pictorial summary encourages a subtle interpretation of the role that every interaction would play in the overall goal of maintaining inventory integrity. The warehouse officer is advised to obtain the delivery note and then add the physical data related to the goods received into the system at the very beginning of the primary workflow and provide this with corroborative evidence. The system as an adjudicator of data integrity will compare the information provided with the expected parameters (purchase orders or delivery lists); it will adjust the inventory ledger only after the verification and hence a greater number of stocks will be counted in the warehouse. In case there is a variance in the number or type classification of items, the system faithfully records the variance and causes a notification to flow to the appropriate administrative or financial custodian. This is also logged in the audit log and thus leaves an untraced trail that can be explored later and eventually reconciled.

4.2.4 Case Diagram Finance

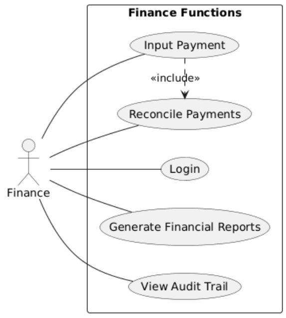

Figure 7. Case Diagram Finance

Source: Processed from study case

Figure 7 shows a case diagram that represents the main activities of Finance module, including payment registration, reconciliation, and the creation of financial reports as well as the access to audit-trails. Operational wise, the Finance unit receives invoices related to the transactional activities, then it adds payment information in the format of amount, method and date after which it connects the payment with the transaction itself with the help of which the transactional status is updated to partial or paid depending on the payment amount. The system introduced the reconciliation process which presents a list of unpaid invoices and inbound payments which facilitate matching files and any variance is allocated as duly logged in order to be remedied. More so, the Finance and Administration departments will be able to produce periodic reports (sales, receivables, and cash-flow statements) and take advantage of the audit trail feature to examine the history of transactions and data alterations.

4.2.5 Case Diagram Admin

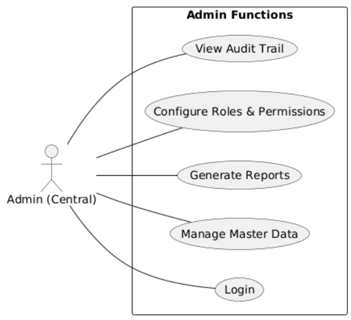

Figure 8. Case Diagram Admin

Source: Processed from study case

In Figure 8, of a case diagram of a role Administrator is provided, with a focus on the significance of master data stewardship, arrangement of access privileges, system set-up process organization, and overall production of analytical reports. The role of administrators is to maintain the main data stores, which consist of product catalogues, storage, and user profile and to maintain the right to create or revoke role-based permissions, such that authorisation schemas can be applied homogenously across modules that are distinct in nature. They also have the mandate to audit and extract audit logs in compliance audits and produce cross-functional reports that will provide managerial stakeholders with information that requires operational and strategic discussions. Each of such maneuvers in an administration is consistently documented through a change-tracking mechanism hence verifying a veracity and tracing of every change.

4.2.6 Case Diagram Customers

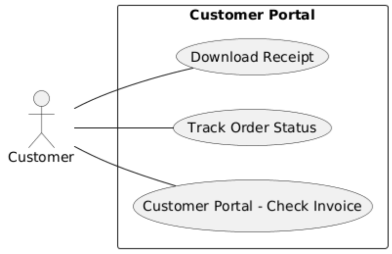

Figure 9. Case Diagram Customers

Source: Processed from study case

Figure 9 shows case diagram of Customer Portal that illustrates a public or restricted interface so that it allows customers to confirm invoices, their delivery status, and download payment proofs. Customers are provided with the ability to access real-time information about bills and the approximate date of goods delivery without entering internal administration modules by providing invoice or tracking numbers. Such a portal improves visibility and reduces the pressure of the internal customer-service processes in a service context, as the customer is made self-sufficient in checking the status of the order and the documents. In addition, any access and information request carried out via the portal is logged, therefore providing needed audit trails and service-analytics initiatives.

4.3 Implications Object Oriented Design (OOD)

4.3.1 Class Diagram

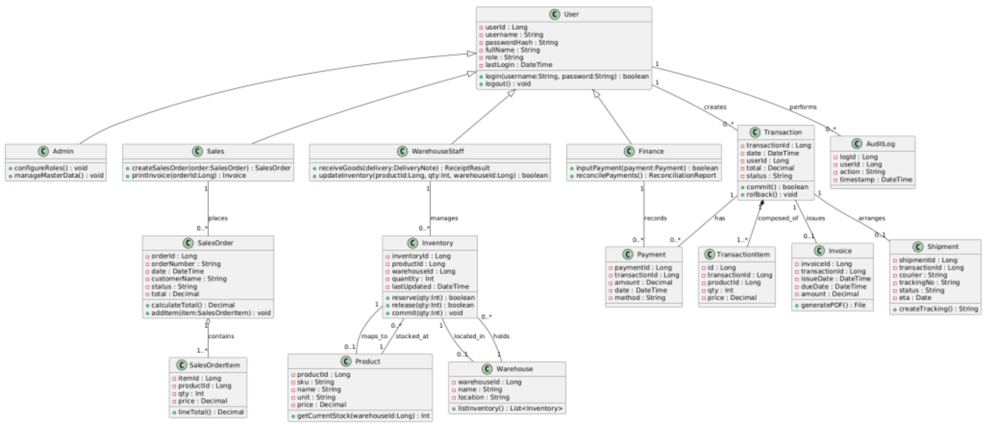

Figure 10. Class Diagram

Source: Processed from study case

As shown at Figure 10, class diagram that defines the object-oriented core class architecture of the IPS system, starting with the user hierarchy (User disaggregated into Admin, Sales, WarehouseStaff, and Finance) and moving on to domain entities, such as Product, Warehouse, Inventory, Reservation and SalesOrder, Transaction, Invoice, Payment, Shipment and DeliveryNote, DiscrepancyRecord, and AuditLog. The diagram also shows important properties of each class- e.g., identifier, name, quantity, status, and timestamp as well as primary functions that comprise service contract e.g reserve, commit, createOrder, validateAgainst order, generatePDF and so on. Further, inter-class associations, aggregations, and compositions that describe the relationships as related to implementation include SalesOrder that aggregates SalesOrderItem; transaction that depends on Transactionitem; Inventory which associates with a product and warehouse; reservation which links Inventory. This story illustrates the benefits of OOD model to create and maintain segregation of responsibilities, ensure data integrity with referential relationships, and support non-functional requirements found in the analysis of PIECES environment such as control with AuditLog, performance with atomic reserve/commit operations, and information with structured bills that simplify the generation of reports.

4.3.2. Sequence Diagram Sales

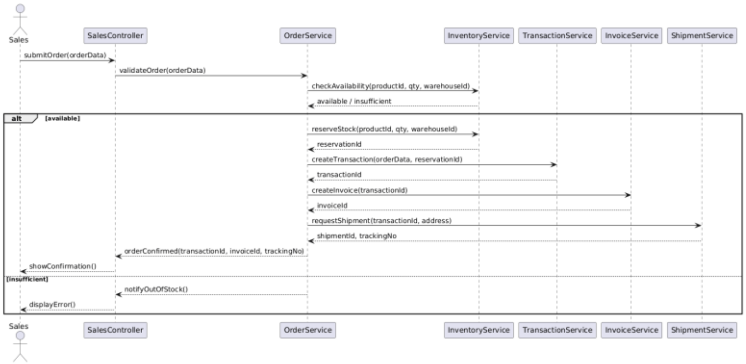

Figure 11. Sequence Diagram Sales

Source: Processed from study case

Figure 11 shows a Sequence Diagram Sales which depicts the time sequence of actions of the messages mutually passed between the system entities: Sales sends an order-creation request to SalesController, which in turn invokes OrderService to verify the data, and in case of a success, the process of creating a Reservation that temporarily holds the ordered quantity is triggered, and after the successful reservation, OrderService invokes TransactionService to create a transaction and InvoiceService to create an invoice, and then a request to ShipmentService is sent to ensure the shipment, which returns This story highlights the sense of emergency on atomicity and rollback schemes in critical points of time to prevent stock and transaction mismatch.

4.3.3 Sequence Diagram Warehouse Staff

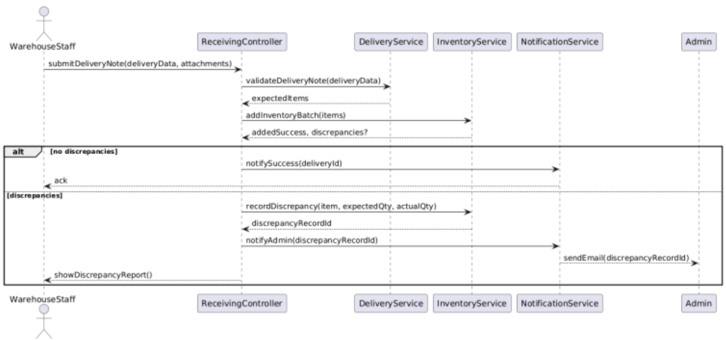

Figure 12. Sequence Diagram Warehouse Staff

Source: Processed from study case

Figure 12 shows a Sequence Diagram in which the Warehouse Staff scenario is described when dealing with the receipt of goods, which captures the sequence of events starting with the submission of a delivery note to the ultimate recording of inventory and, in case of a problem, a discrepancy notification is issued. Starting from the Warehouse Staff input, the deliveryData and related attachments are sent to the ReceivingController, who requests the DeliveryService to authenticate the list of received goods, and when it authenticates the ability of the inventory, the Receiving Controller requests the InventoryService to add to the inventory according to the batch received and registers the resultant state. In case some deviation is identified between the projected inventory and the actual state, the diagram outlines the procedures involved in the creation of a DiscrepancyRecord and invoking of the Notificationservice to either notify the administrative or finance department and to produce an audit log that would be used to investigate the situation further. This story highlights the fact that receiving process should utilize human judgment by use of manual verification as it automates data updates that has the capacity to reduce input errors and hasten reconciliation.

4.3.5 Activity Diagram

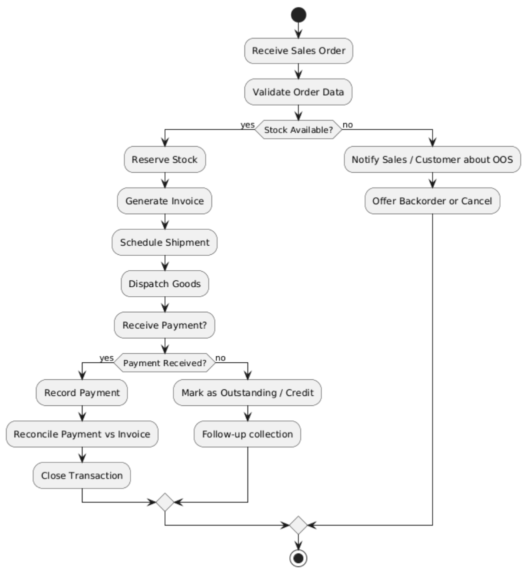

Figure 13. Activity Diagram

Source: Processed from study case

Figure 13 represents an activity diagram, which provides a chronologically arranged business flow within the Order-to-Cash, and it starts with entrance of order and completes with closing of transaction. The steps start with a sale order being received and further validation of the data that has been supplied. It then goes on to determine the availability of stocks and on a condition of sufficient inventory, make a reservation. After that, the invoice will be created and the delivery will be planned. The shipping process would be completed until possession of the goods to the consignee after which the finance department would receive payment and match it with the invoice. This is ended by the closing of the transaction or the acculturation of any outstanding balance in case payment is not done. The flows that are also mapped in the diagram include notifying sales or customers of stock being run out, back order or cancellation options, and follow up on outstanding receivables. This story explains how to divide roles in actors, the control nodes to be considered in the system, such as validation, authorization, notification, and the efficiency goals to be achieved in the operational improvement.

4.3.6 State Diagram

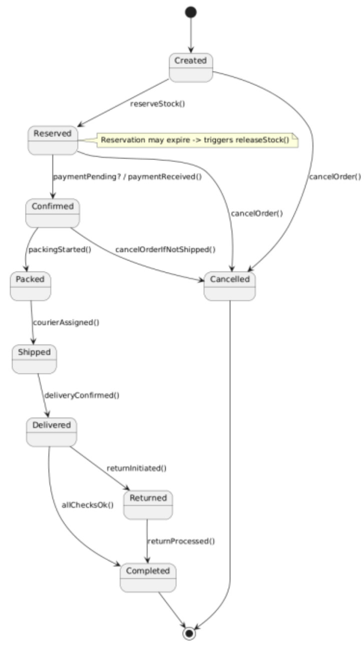

Figure 14. State Diagram

Source: Processed from study case

The state diagram of Figure 14 represents a model of SalesOrder object lifecycle in terms of well-defined transitional states. An order will be initially in the Created state, then move through the Reserved state when stock is occupied, then move through the Confirmed state when the necessary payment or verification is received, then move through the Packed and Shipped states as the logistics are carried through and finally move through the Delivered and Completed states when all the necessary validation is complete. The scheme also represents such alternative transitions as the Cancelled state can be arrived at through various antecedent states (Created, Reserved, Confirmed) and the Returned path which can lead to returns and new breed of stock reconditioning. This state narrative focuses on the time aspect of the process, the business policy on when the activities on an entity can cause inventory changes, releasing a reservation on cancellation, committing inventory on shipment, and how expiry or reservation timeouts can go to the available pool. It also points to the consistency of data and support of the transversal functions, like notifications and auditing, that this state machine supports.

4.4 Database Design and Technical Output

Database paradigm is a relational one with a high value on referential integrity, thorough data normalization, and support of atomic transactions using Entity Relactionship Diagram (ERD). The schema includes master, operational and transactional objects correlated by a foreign key, which make it possible to track the data process in a logical and coherent way since the time a certain order is placed and reservation of stock is made all the way to the moment when payments are logged and the history of further activities is recorded.

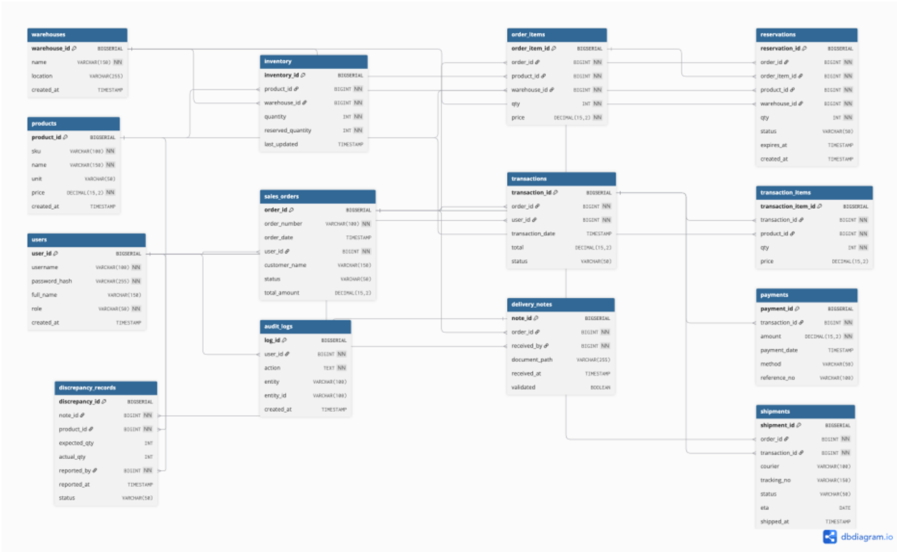

Figure 15. Entity Relactionship Diagram (ERD)

Source: Processed from study case

As shown by Figure 15, the suggested database design is constructed to ensure referential integrity, offer powerful reporting flexibility in addition to offering to supply atomic transaction capabilities of the most critical activities, including stock reservations and transaction creation. The relational schema follows the principles of normalization with a rigor that may be defined as 3NF where the main entities are concerned, therefore, the relational database is as less redundant as it can be. The main tables include the user, products, warehouses, inventory, transactions, transactionitems, payments, shipments, reservations, deliverynotes, discrepancyrecord, and auditlogs. Primary keys It is recommended to index numeric types depending upon the scale of the deployment; commonly used columns in searches, such as sku, warehouseid, transactionid, and order number, are also indexed to increase the speed of reads in dashboards and reports.

The database transaction engine (Begin/Commit/Rollback) is designisely crafted as a critical process where it is guaranteed that the reserve, create- transaction, and commit sequence is atomic in nature which prevents the occurrence of race conditions which otherwise can result in overselling. Data-logic level also provides status and timestamp columns to each transactional entity enabling extensive auditing, thorough reconciliation, and sound time-of-event analytics. Technically, what is out of scope of the DDL (Create Table) script, but is within the scope of the deliverables, are seed-data scripts of example account and product/warehouse configuration, simple migration scripts, and a battery of sample reporting queries, e.g. stock per warehouse, transaction history per customer, and a reconciliation report, which are physically actually technical attachments. All the schema designs and sample codes are synchronized with the design artifact (ERD), through which the developers and DBAs can effortlessly transform the conceptual model into a technically viable implementation.

4.5 Interface Design and Operational Processes

The fundamental concern of the current interface design revolves around functional exigencies; it is the support of accurate data entry in the field, the speed of administrative processes, and the ability to provide minimal information displays to managers control. The main pages include a role based authentication system, a per role dashboard with salient indicators which include low stock notices, pending payment notices, delivery discrepancy notices, an order creation module which has a warehouse picker and quantity validation and a goods receipt form which accepts delivery notes and photo uploads and a payment reconciliation page which has date filtered and status filtered controls. The interface is designed to support mobile input devices and has strict validation controls- numerical formatting of quantities, SKU violations, maximum uploads and confirmation required on any operations that change the stock amounts. Under the integrated workflow, the availability-check result and outbound reservation identifier are shown on the user interface when an order is either made by the sales function creating an order, the receipt form will then verify the purchase order or originating order accomplished with the entry of any variance as a discrepancy record and the finance functionalities match the payment to the correct invoice to gain a reconciliation. Furthermore, sensitive operations are done by use of tooltips as well as dual confirmations, therefore, reducing the human factor and speeding up the recording and audit cycle.

4.6 Security, Control and Implementation Priorities

The architecture of the building is made up of security and control as the skeletons. User access is built on an authentication system which uses password hashing with bcrypt used along with stateless authentication through JWT or secure cookies. Role-based access control (RBAC) defines the relationships of privileges in administrative, sales, warehouse, finance, and customer functions. Precisely, an audit trail is maintained and the actor, action and time are recorded on each of the operations. The implementation protection against an injection attack uses an object-relational mapper (ORM) and parameterized queries, complemented by intensive input sanitization and server validation. Small measures like rate limiting, extensive access logging and any anomaly detection are recommended and the transport layer is encrypted with Hypertext Transfer Protocol Secure (HTTPS).

Operation controls are also of paramount importance. To keep inventory unoccupied indefinitely a reservation expiry mechanism will prevent this form of inventory and detect conflicts with stock items by double-booking of stock items. Administrators still have a discretion to override reservations whose individual override will be recorded in the audit trail. The roadmap of development is systematically graded: the minimum viable product is focused on the verification of identity, real-time inventory management, and the system of reservation. The later stages bring on board sales processes, invoicing and shipment processing; payment and reconciliation system and finally a customer portal; followed by advanced reporting, performance optimization and external systems integration. This gradual implementation will provide a speedy deployment and limit the chances of disruptive implementation.

4.7 Impact Evaluation and Critical Discussion

In line with the theory of change, the integrated system is expected to improve the accuracy of inventory, speed up reconciliations, and increase the visibility of operations in the process of accelerating decision-making. The pilot measures include: the frequency of inventory variances per month, the average time taken to reconcile payment, order-to-shipment latency, and customer satisfaction with the tracking and invoicing process. On the qualitative level, the system will reduce re-work caused by manual data entry, as all operations are well-documented. However, the effectiveness of the system depends on the original data quality and a high adherence to standard operating procedures; the infrastructure of the warehouse connectivity also poses a limitation and it is necessary to implement offline-first or batch-synchronization approaches in challenging locations. Organizational changes require thorough training and solid change management without which an essential user not only may ignore such features like stock reservations but also may overlook the opportunity to do it. As a result, evaluation has to include both technical quantitative metrics and qualitative human-related aspects.

4.8 Limitations of Results and Further Recommendations

Such findings are abstract and are pegged on a case study document, and hence there are limitations in using them in a normative way without subjecting them to strict field testing. Its major limitations are the use of illustrative data, the non-existence of the prototype that was confirmed empirically, and the assumption of perfect infrastructural conditions (e.g. real-time connectivity). Recommendations: pilot in an appropriate warehouse to get empirical data and to do a quantitative pre-post assessment of the key performance indicators; perform load testing and transaction-integrity testing on a production RDBMS; design off-line fallback mechanism that would synchronise the data after connectivity is reestablished. On the organizational level, an overall master-data policy, the multi-level training programme, and a change-governance framework, which would involve the most important stakeholders, is invaluable. Future studies would explore automated connections with payment gateways and freight forwarders and how predictive analytics can be used to optimise stock projections.

5. DISCUSSION, CONCLUSION AND RECOMMENDATION

5.1 Discussion

The current research combined an object-oriented system design and PIECES-guided diagnostic in order to reduce the longtime inventory and reconciliation issues at PT Asia Persada Nusantara. The empirical data displays the following interrelations: (a) unexplained stock gaps clumped together in a few warehouses, (b) regular manual over-rides and ad-hoc reconciliation process, (c) late payment matching and invoice mismatch, (d) ill visibility of user activities across the current workflow. These notes and enhance the PIECES diagnosis expressed in Chapter 4 and shed light on the deficiencies across several dimensions: performance (long reconciliation time), information (varied source of data), control (inadequate audit and authorization), efficiency (duplication of entries), economy (employed labor to rework), and service (customers complain and there are delays).

These findings have three substantive insights which are generated through a theoretical interpretation process. First, the operational failure is not purely technical, but socio-technical: poor-quality data and discontinuous procedures are increasing technical limitations, which creates repetitive business issues. This point of view supports the prior literature that emphasizes the need to align technical architectures to user processes and driving incentives. Second, design options entailing the implantation of process controls, including the reservation expiration policy and required received checks, will likely limit the risk of manual errors and, consequently, enhance accountability; the expectation is confirmed by artefacts of the prototype presented in Chapter 4. Third, the stage-based MVP implementation with an underlying set of KPIs is a application that will help to manage organisational risk as well as provide measurable positive results; baseline pilot metrics will be essential to determine the contribution of separate modules (reservation logic, audit logs, UI restrictions).

These findings are comparatively analyzed with available literature, and they all point towards the same direction that has been elaborated in the literature that operational benefits are greater when the systems consider process redesign with technical constraints and not the architecture taken independently. Unlike a lot of the literature which assumes dependable connectivity and centralized authority, as is shown in this case study, intermittent connectivity and decentralized operational responsibility necessitates design styles that uphold local validation, asynchronizing consistency, and minimal synchronization patterns.

5.2 Practical Implications and Concrete Recommendations

The recommendations that follow can be considered as concrete, prioritized and actionable. The most significant party in charge, an estimated schedule or ranking is listed in each item, and demonstrative criteria of the success are quantified: a) Pilot the MVP in one representative warehouse; b) Enforce role-based access and mandatory audit logging; c) Implement reservation logic with explicit expiry and notification; d) Standardize goods receipt verification with digital evidence; e) Define KPI dashboard and weekly monitoring cadence; f) Institutionalize training and change management; g) Prepare rollout criteria and contingency planning

Each recommendation is small, testable, and bounded to specific actors and metrics. Avoid generic language; these steps can be operationalized by the company in the pilot phase and then adapted for scale.

5.3 Broader social and organizational implications

These findings go beyond the operation efficiency. The improvement of inventory accuracy and reconciliation will potentially reduce the number of hours spent by employees on re-work and could allow employees to participate in tasks that have more value and could potentially result in job satisfaction. Through alleviating invoice/payment mismatch, cash-flow velocity may be increased, which will help to improve relations with vendors and add stability to the local supply chains. Fair and meaningful supply at the community level and minimization of risks of stockout conditions help to protect consumer welfare in both semi-urban and rural areas. However, the managers should be made aware of potential tradeoffs: the increased vigilance can cause the perception of the surveillance and ensure the necessity to communicate and manage the change intentionally to prevent this risk of the negative impact on morale.

5.4 Limitation

This question is restricted by a number of constraints limiting the extrapolation of the findings and the causal inference of the results: (1) Single-case design: the factual background is rooted in a single corporation environment which may not be easily generalized to firms of varying sizes, governance systems or connectivity systems; (2) Crosssectional scope of pilots: without the longitudinal follow-up, short-term effects of the pilot can be taken as the temporary learning curves but not to reflect long-term changes; (3) Constraints of data: Historical records of transactions may be incomplete or include some noise in them and some KPIs are dependent on the integrity of the information recorded in metadata; measurement error is therefore a real issue; (4) Attribution issues: the enhancement that may be realized in the pilot can be attributable to other organizational changes that are also underway; attribution will require a harsh design, including control sites or comparison sites.

These shortcomings do not reject the working value of the research but it merits cautious generalization of the results. The next section identifies specific research plans that will overcome such inadequacies.

5.5 Future research directions

The limitations mentioned above should be overcome in future studies which should be out-witted by carefully designed studies: (1) Multi-location comparative pilots involving more than one warehouse or similar company to probe into the external validity; (2) Longitudinal studies which measure the extent of sustainability of the gains and adoption pattern of the user; (3) Pilots of quasi-experiments or matches of the controls to strengthen causality of attribution; (4) Socio-technical exploration such as incentive structure, informal practices and organizational culture which mediates the adoption of technology; (5) Technical research of synchronization techniques of intermittent connection and lightweight conflict-resolution techniques befitting regional trading networks.

All these avenues are completely viable in the terms and conditions of existing operations and will bring about significant evidence to support mass deployments.

5.6 Conclusion

This paper depicts the operationalisation of a PIECES-based diagnosis on design artefacts and a phased MVP roadmap to solve the inventory and reconciliation challenges that have been plaguing the operations of a multi-warehouse regional trading company. The dreamed-up Integrated Inventory Information System highlights the logic of reservation, role-based controls, auditing, and pragmatic UI restrictions, whose functions are intended to curb the errors and increase responsibility. Implementing the pilot based on strictly spelled KPIs is the rational forwarding step into the validation of the design on an empirical level. The combination of diagnosis, design, and pilot evaluation makes this research provide a viable outline to practitioners and methodologically inform the discussion about the use of PIECES in order to manage system design in resource-constrained, distributed operation settings.

Ethical Approval

This study did not require ethical approval, as it used operational company data and case study with lecturer Information System Islamic University of Indragiri permission with PT. Asia Persada Nusantara.

Informed Consent Statement

This research did not require informed consent.

Author's Contributions

Conceptualization, UAS; methodology, UAS, DI; validation, D.I; formal analysis, UAS., DI; resources, DI; writing – original draft preparation, UAS; writing – review and editing, UAS., and DI.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Data Availability Statement

The data presented in this study are available on request from the corresponding author due to privacy reasons.

Funding

This research received no external funding. All costs associated with the preparation and publication of this article were fully supported by the parents of Uya Asy Syuura Anandri (ORCID: 0009-0000-2385-7400).

Notes on Contributors

Uya Asy Syuura Anandri
https://orcid.org/0009-0000-2385-7400
Uya Asy Syuura Anandri is an under graduate student undertaking an Information Systems Program at Universitas Islam Indragiri. Expertise areas include database design, such as the development of Entity Relationship Diagrams (ERD), descriptive statistics analysis and implementation of information systems in microenterprises, small- and medium-sized enterprises (UMKM) and Implementation Artificial Intelligence (AI) and Business Intelligence. Contributions to research include case studies and practice reports on the local UMKM business models in Tembilahan, and the creation and deployment of information systems, based on ERD, to small enterprises. Data on the audit and deployment of information-systems in community-related settings is found in publications and conference proceedings. Uya is still actively involved in campus research work and other community service activities with a vision of changing the academic findings into local economic empowerment.

Ilyas
Ilyas is Lecturer in the Information System, Faculty Engineering and Computer Science, Islamic University of Indragiri (UNISI). He has a long record of research and publication in information systems and computing, including topics such as decision support systems, web-based information systems, and system analysis applied to organizational and community contexts. Ilyas has published multiple articles in national scientific journals and has been actively involved in research related to information system design and implementation.

REFERENCES

Alfassa, A. I., Sudrajat, S., & Marwasta, D. (2023). Development of official statistics models for analysis of population sectoral data in Indragiri Hilir Regency. *E3S Web of Conferences*, *468*, 06007.

Alles, M. G., Kogan, A., & Vasarhelyi, M. A. (2018). Putting continuous auditing theory into practice: Lessons from two pilot implementations. *Journal of Information Systems*, *32*(1), 53–73. https://doi.org/10.2308/isys-51763

Anandri, U. A. S. (2025). Business plan and financial feasibility study of "Prata Bubuhan" breakfast UMKM stall in Indragiri Hilir Regency. *Priviet Social Sciences Journal*, *5*(11), 388–399. https://doi.org/10.55942/pssj.v5i11.746

Anandri, U. A. S., & Arkan, M. N. (2025). Pemanfaatan Power BI Untuk Visualisasi Data Penjualan Produk Pada Perusahaan Percetakan. *Jurnal Sistem Informasi (TEKNOFILE)*, *3*(10), 716–721. https://jurnal.nawansa.com/index.php/teknofile/article/view/562

Anandri, U. A. S., Arkan, Mhd. N., Abdullah, & Ilyas. (2025). Implementasi dan Pengembangan Aplikasi Point Of Sales Berbasis Website pada UMKM RitelShop Tembilahan. *Jurnal Pengabdian Masyarakat (ABDIMAS)*, *3*(7), 304–314. https://jurnal.nawansa.com/index.php/abdimas/article/view/751

Anandri, U. A. S., Nabil Arkan, M., Risky, M., Elma Sudiarti, Y., Isya Alfassa, A., & Rachman, A. (2025). Analisis Statistika Deskriptif Data Mahasiswa Sistem Informasi B 2024 Universitas Islam Indragiri. *TEKNOFILE: Jurnal Sistem Informasi*, *3*(6), 422–432. https://jurnal.nawansa.com/index.php/teknofile/article/view/619

Anandri, U. A. S., & Prasetyo, D. Y. (2025). Analisis Model Bisnis Inovatif Kue Pancung Keliling Untuk Pemberdayaan UMKM Tembilahan. *ABDIMAS: Jurnal Pengabdian Masyarakat*, *3*(6), 204–212. https://jurnal.nawansa.com/index.php/abdimas/article/view/629

Arkan, Mhd. N., & Anandri, U. A. S. (2025). Perancangan Entity-Relationship Diagram (ERD) Menggunakan SQL Server Database Pada Platform Tautanhalaman Web Lynk.Id. *TEKNOFILE: Jurnal Sistem Informasi*, *4*(3), 221–227. https://jurnal.nawansa.com/index.php/teknofile/article/view/411

Aswintama, P., Haryanto, E., & Setyawan, R. A. (2025). Implementation of role-based access control, multi tenancy and audit logging in a single sign-on system. *Jurnal Mandiri IT*, *14*(1), 119–128. https://doi.org/10.35335/mandiri.v14i1.441

Atieh, A. M., Kaylani, H., Al-Abdallat, Y., Qaderi, A., Al-Tamimi, O., & Al-Faouri, N. (2016). Performance improvement of inventory management system processes by an automated warehouse management system. *Procedia CIRP*, *41*, 568–572. https://doi.org/10.1016/j.procir.2015.12.122

Booch, G., Rumbaugh, J., & Jacobson, I. (2007). *The Unified Modeling Language User Guide*. Addison-Wesley.

Dobing, B., & Parsons, J. (2006). How UML is used. *Communications of the ACM*, *49*(5), 109–113. https://doi.org/10.1145/1125944.1125949

Gunasekaran, A., Subramanian, N., & Rahman, S. (2017). Supply chain resilience: Role of complexities and strategies. *International Journal of Production Research*, *55*(22), 6809–6819. https://doi.org/10.1080/00207543.2017.1362936

Hendricks, K. B., Singhal, V. R., & Stratman, J. K. (2007). The impact of enterprise systems on corporate performance. *Journal of Operations Management*, *25*(1), 65–82. https://doi.org/10.1016/j.jom.2006.05.002

Huang, Z., Kim, S. H., & Sadjadi, S. J. (2019). Order allocation and inventory reservation in multi-channel distribution systems. *International Journal of Production Economics*, *216*, 47–60. https://doi.org/10.1016/j.ijpe.2019.04.012

Ilyas, Putri, A. K., & Pahrianti, M. (2024). *Sistem Pendukung Keputusan & Implementasi Metode* (M. H. Maruapey, Ed.; 1st ed., Vol. 1). PENERBIT KBM INDONESIA.

Kembro, J., Selviaridis, K., & Naslund, D. (2017). Theoretical perspectives on information sharing in supply chains. *Supply Chain Management: An International Journal*, *22*(5), 401–417. https://doi.org/10.1108/SCM-01-2017-0006

Lopulalan, P. M., Suhartono, R. H. S., Bakri, B., Karim, A., & Anandri, U. A. S. (2025). Reconstructing a Sustainable Quality Management System in Higher Education through an Integrative Approach between Academic Audits and Organizational Reflection. *Journal of Educational Analytics*, *4*(3), 675–690. https://doi.org/10.55927/jeda.v4i3.362

Ouyang, F., & Jiao, P. (2021). Artificial intelligence in education: The three paradigms. *Computers and Education: Artificial Intelligence*. https://doi.org/10.1016/j.caeai.2021.100020

Prasetyo, D. Y., Sudiarti, Y. E., Anandri, U. A. S., Risky, M., & Ihwan, K. (2025). Website E-Commerce Pada Toko Oska Cake. *JUTI UNISI*, *9*(2), 31–40. https://doi.org/10.32520/juti.v9i2.4834

Rai, A., Patnayakuni, R., & Patnayakuni, N. (2002). Technology investment and business performance. *Information Systems Research*, *13*(1), 3–23. https://doi.org/10.1287/isre.13.1.3.97

Ramadhan, F. A., & Mansur, A. (2024). Bibliometric Analysis and Data Visualization: Business Intelligence in Digitalization of Supply Chain Management in Healthcare Sector. *Prozima: Productivity, Optimization, and Manufacturing System*, *8*(2), 115–131. https://doi.org/10.21070/prozima.v8i2.1728

Rianto, B., Jalil, M., Sandi Rais, M., & Bindas, A. (2023). The Development Of Enterprise Architecture Business CRM At Pt. Capella Dinamik Nusantara With EAP Methodology. *Jurnal Perangkat Lunak*, *5*(2), 65–73.

Sommerville, I. (2016). *Software Engineering*. Pearson.

Vasarhelyi, M. A., Kogan, A., & Tuttle, B. M. (2015). Big data in accounting. *Accounting Horizons*, *29*(2), 381–396. https://doi.org/10.2308/acch-51068

Whitten, J. L., & Bentley, L. D. (2007). *Systems Analysis and Design Methods*. McGraw-Hill.