International Journal Software Engineering and Computer Science (IJSECS)
6 (3), 2026, 1099-1110
Published Online December 2026 in IJSECS (http://www.journal.lembagakita.com/index.php/ijsecs)
P-ISSN: 2776-4869, E-ISSN: 2776-3242. DOI: https://doi.org/10.35870/ijsecs.v6i3.8249.

RESEARCH ARTICLE Open Access

Implementation of a Web-Based Inventory Monitoring System at Optik Sanjaya Depok

Axel Rajaliu Chaniago [1*], **Frencis Matheos Sarimole** [2]

[1*,2] Department of Informatics Engineering, Faculty of Computer Technology, Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika, East Jakarta City, Special Capital Region of Jakarta, Indonesia.

Article History:

Received: July 29, 2026.
Revised: July 30, 2026.
Accepted: August 27, 2026.
Available online: September 20, 2026.
Published: December 1, 2026.

***Corresponding Author:**
axelrajaliuc@gmail.com

Abstract: Optik Sanjaya Depok, a micro, small, and medium enterprise (MSME) engaged in eyewear retail, still recorded its eyeglasses stock manually in handwritten ledgers, which frequently resulted in recording errors, delayed stock updates, difficulty in determining available items, and time-consuming report preparation. This study aimed to design and implement a web-based stock monitoring system to establish more structured, efficient, and traceable inventory management for the store. The system was developed using the Waterfall software development life cycle, covering requirement analysis, Unified Modeling Language (UML) and Entity-Relationship Diagram (ERD) design, implementation using PHP Native, MySQL, and Bootstrap, and testing. The system provides role-separated access for the Administrator and the Warehouse Admin, a summary dashboard, user management, item and item-category data management, incoming-stock and outgoing-stock recording that automatically updates item quantities, real-time stock monitoring, stock reporting with Excel export, and DataTables-based search, filter, sort, and pagination. Black box testing was applied to ten core features, and every test scenario produced the expected result. User Acceptance Testing using a five-point Likert-scale questionnaire completed by two respondents consisting of the owner and warehouse staff of Optik Sanjaya Depok yielded average scores between 4.7 and 4.9 (overall average 4.82). The system replaced the manual recording process and supported centralized, consistent stock recording; the reduction of stock-data errors was not measured quantitatively in this study and is therefore not claimed as a research finding. These findings indicate that a role-based, web-based stock monitoring system is a feasible initial step in the digital transformation of small-scale optical retail businesses.

Keywords: Stock Monitoring System; Web-Based Information System; Inventory Management; Waterfall Method; Optical Retail MSME.

1. Introduction

Digital transformation continues to reshape how micro, small, and medium enterprises (MSMEs) manage their daily operations, pushing many business owners away from manual, paper-based recording toward integrated, web-based information systems that improve accuracy and operational efficiency (Ridwansyah *et al.*, 2023; Aprilisa & Aulia, 2024). Information technology allows MSME owners to manage inventory, transactions, and reporting with a level of precision and speed that manual bookkeeping cannot match (Damanik *et al.*, 2021; Wau, 2022). MSMEs contribute significantly to local economic activity and employment, yet a large number still rely on limited digital infrastructure, particularly in inventory and stock management (Suhada *et al.*, 2026). Optik Sanjaya Depok, an optical retail business selling eyeglasses, frames, and related products, represents one such case. Prior to this study, the store recorded incoming and outgoing eyeglasses stock manually in handwritten ledgers, without a centralized database. This manual approach caused frequent recording errors, delays in updating stock quantities after transactions, difficulty for staff to determine which items were currently available, and considerable time spent compiling stock reports for the owner. The absence of a structured recording mechanism also made it difficult to track item categories, identify slow-moving or near-depleted stock, and maintain a reliable audit trail of stock movement. Previous studies on web-based inventory and stock information systems have addressed these issues only partially, and the specific combination of functions required by a small optical retail store has not yet been reported. General inventory studies such as Damanik *et al.* (2021), Wau (2022), and Ananda *et al.* (2024) validate the Waterfall model for this class of system, but their scope is a single administrative user rather than access that is separated by role. Studies conducted in a warehouse context, such as Susanto and Purnomo (2023) and Hikmal *et al.* (2025), concentrate on material and warehouse stock administration rather than on the item categories handled in a retail eyewear store. In the optical retail domain itself, Imam (2020), Yofena (2022), and Indarto (2022) developed information systems for eyeglasses stores, but their focus is point-of-sale and customer-facing sales transactions, so internal incoming and outgoing stock movement is not treated as a monitored process. The study closest to the present work in this domain, Alfarizi and Priyadi (2024), does develop a web-based eyeglasses inventory system, but it was built with Extreme Programming and reports neither role-separated access nor an exportable stock report. Studies that do provide structured incoming- and outgoing-stock modules, such as Faturrachman *et al.* (2025) and Ramadhan *et al.* (2025), report those modules without an accompanying real-time monitoring dashboard for the owner, whereas monitoring-oriented studies such as Setiawan *et al.* (2021), Tolampi and Maria (2024), and Suhada *et al.* (2026) present stock monitoring without role-based access control. Finally, exportable reporting that allows the owner to obtain a stock recapitulation without manual re-typing is seldom reported as an integral feature; Fahrizal and Azhar (2025) and Fathoni and Kustiyono (2025) discuss reporting efficiency but not a spreadsheet export mechanism combined with client-side search, filter, sort, and pagination. Therefore, this study addresses the combination of role-separated access, structured incoming and outgoing stock modules, real-time stock monitoring, and Excel-based reporting within a single system for a micro-scale optical retail business. To address this gap, this study aims to: (1) design and deploy a web-based stock monitoring system for Optik Sanjaya Depok; (2) implement multi-user login that separates Administrator and Warehouse Admin roles; (3) build item and item-category data management modules; (4) implement incoming-stock and outgoing-stock recording modules that automatically update inventory quantities; (5) provide a real-time stock monitoring module and a monitoring dashboard for the owner; and (6) provide a stock reporting module supporting search, filter, sort, and pagination through DataTables, together with an Excel export feature. The scope of this research is limited to internal inventory and stock management functions. The system focuses on recording and monitoring stock movement rather than processing customer-facing retail transactions or point-of-sale checkout; it does not integrate barcode or RFID scanning hardware; and the Excel export feature is intended for reporting purposes rather than as a full business-intelligence analytics module. These boundaries keep the research focused on a technically achievable and verifiable prototype within the available development time.

2. Related Work

A number of prior studies have examined the development of web-based inventory, stock monitoring, and warehouse information systems for small businesses and institutions, as well as the testing methods commonly used to validate such systems. Table 1 summarizes the studies most relevant to this research and clarifies how each relates to the present work.

Table 1. Summary of Related Studies

No	Author(s) & Year	Research Focus	Relation to This Study
1	Damanik, Meilano & Wr (2021)	Development of a goods-inventory information system using the Waterfall model	Confirms the Waterfall model as a suitable SDLC for inventory systems; access is not separated by role
2	Wau (2022)	Web-based warehouse inventory information system developed with the Waterfall model	Warehouse-oriented stock recording; monitoring is not provided as a dedicated real-time module
3	Setiawan *et al.* (2021)	Web-based stock monitoring application at PT Intermetal Indo Mekanika	Closest reference for the stock monitoring concept, but in a manufacturing rather than optical retail context
4	Ridwansyah, Rifqie & Nuridayanti (2023)	Web-based store inventory system for a costume-rental MSME	Shows the relevance of web-based inventory systems for MSMEs; the item characteristics differ from eyewear products
5	Ananda, Butsianto & Sulaeman (2024)	Application of the Waterfall method to a website-based inventory information system	Methodological reference for Waterfall-based inventory development
6	Aprilisa & Aulia (2024)	Prototype method applied to a web-based goods-inventory information system	Alternative SDLC reference that supports the choice of a structured development model
7	Faturrachman *et al.* (2025)	Web-based stock management application with incoming-goods, outgoing-goods, and goods-request modules	Closest reference for the incoming/outgoing stock modules, but without a monitoring dashboard for the owner
8	Ramadhan *et al.* (2025)	Web-based stock management information system for a retail store using the Waterfall model	Comparable module structure in general retail; the optical retail context is not covered
9	Fathoni & Kustiyono (2025)	Inventory information system using the Waterfall model to improve stock-management efficiency	Supports the efficiency objective of this study; reporting is not exported to a spreadsheet format
10	Fahrizal & Azhar (2025)	Web-based raw-material inventory reporting system using the Waterfall model	Reporting-oriented reference; the reporting object is raw material rather than retail eyewear items
11	Hikmal, Yudertha & Ramadhani (2025)	Web-based material stock management system in the PLN ULP Telanaipura warehouse	Institutional warehouse context; role separation for a two-person MSME is not addressed
12	Susanto & Purnomo (2023)	Web-based warehouse information system built with the Laravel framework	Framework-based warehouse reference; the present study uses PHP Native for a lightweight deployment
13	Suhada *et al.* (2026)	Web-based stock monitoring system to improve inventory management efficiency in an MSME	Directly relevant MSME stock-monitoring reference; the case study is not an optical retail store
14	Adha (2022)	Web-based goods-inventory system using the Waterfall model in a government transport agency	Public-sector inventory reference; the recording flow is administrative rather than commercial
15	Tolampi & Maria (2024)	Information system for tracking and monitoring the position of goods using the Waterfall model	Monitoring reference oriented towards the position of goods rather than the quantity of stock
16	Imam (2020)	Web-based eyeglasses sales information system at Indo Optikal, East Jakarta	Optical retail reference focused on sales transactions rather than internal stock monitoring
17	Yofena (2022)	Web-based eyeglasses sales information system at Optik Yonotrend	Optical retail reference limited to the sales side of the business

18	Indarto (2022)	Android-based eyeglasses sales information system at Optik Kunanti	Optical retail reference on a mobile platform, without internal stock-movement recording
19	Alfarizi & Priyadi (2024)	Web-based eyeglasses inventory information system for an optical store, developed with Extreme Programming	Closest prior work in the optical inventory domain; uses a different development method and reports neither role-separated access nor an exportable stock report
20	Shadiq, Safei & Ratu Loly (2021)	Black box testing applied to an office-vehicle loan application	Methodological reference for the black box testing procedure adopted in Section 3.5
21	Aliyah, Hartono & Muin (2025)	User Acceptance Testing applied to a financial and goods-inventory information system	Methodological reference for the Likert-scale UAT adopted in Section 3.5

The reviewed studies tend to treat inventory recording, warehouse administration, stock monitoring, and optical retail sales as separate concerns. General inventory studies (Ananda *et al.*, 2024; Damanik *et al.*, 2021; Wau, 2022) validate the Waterfall model for this type of system but do not separate access by role; studies that provide incoming- and outgoing-stock modules (Faturrachman *et al.*, 2025; Ramadhan *et al.*, 2025) do not combine them with a monitoring dashboard; monitoring-oriented studies (Setiawan *et al.*, 2021; Suhada *et al.*, 2026; Tolampi & Maria, 2024) present monitoring as a stand-alone concern; and studies in the optical retail domain (Imam, 2020; Indarto, 2022; Yofena, 2022) concentrate on sales rather than on internal stock movement. Table 2 summarizes the resulting research gap.

Table 2. Research Gap and Novelty Analysis

Aspect	Previous Studies	This Study
Role-separated multi-user access (Administrator and Warehouse Admin)	Rarely reported; most reviewed inventory systems operate under a single administrative account	Implemented and verified in testing
Incoming- and outgoing-stock modules that update item quantities automatically	Reported in part of the reviewed studies	Implemented as the core of the recording workflow
Real-time stock monitoring supported by a summary dashboard	Usually developed as a separate monitoring application	Integrated within the same system and database
Stock reporting with Excel export	Seldom reported as an integral feature of the system	Implemented in the stock report module
DataTables-based search, filter, sort, and pagination	Not explicitly reported in the reviewed studies	Applied to all data tables in the system
Optical retail (eyewear) case study	Existing optical retail studies focus on sales and point-of-sale functions	Focused on internal stock monitoring at Optik Sanjaya Depok

The novelty of this research lies in combining, within a single lightweight web application built for a micro-scale optical retail store, four elements that the reviewed literature reports separately: role-separated access for the Administrator and the Warehouse Admin, incoming- and outgoing-stock recording that automatically updates item quantities, real-time stock monitoring supported by a summary dashboard, and stock reporting with Excel export and DataTables-based search, filter, sort, and pagination. The contribution of this study is therefore not a new algorithm or technique, but the demonstration that this combination can be implemented, tested, and accepted in the operational environment of a small optical retail business.

3. Methodology

3.1 Research Method

This research adopts an applied research approach aimed at developing a web-based information system prototype to digitalize the stock-recording workflow of Optik Sanjaya Depok. The Software Development Life Cycle (SDLC) used is the Waterfall model, chosen for its rigid, sequential structure in which each phase must be validated before the next phase begins. The five phases implemented were: (1) Requirement Analysis, in which functional and non-functional needs were elicited from observation and interviews regarding the store's manual stock-recording process; (2) System Design, in which Unified Modeling Language (UML) diagrams and

an Entity-Relationship Diagram (ERD) were produced; (3) Implementation, in which the front-end and back-end were coded using PHP Native and MySQL; (4) Testing, consisting of Black Box Testing and User Acceptance Testing (UAT); and (5) Maintenance, covering bug fixing and minor refactoring after deployment.

3.2 Requirement Analysis

Functional requirements defined for the system include: multi-user login separating the Administrator and Warehouse Admin roles; a summary dashboard; user account management; item (eyeglasses/frame) data management; item-category management; incoming-stock recording that automatically increases item quantities; outgoing-stock recording that automatically decreases item quantities; real-time stock monitoring; stock report generation with Excel export; and full Create, Read, Update, Delete (CRUD) operations for item, category, and transaction data, supported by DataTables-based search, filter, sort, and pagination. Non-functional requirements cover accessibility, so that the system can be accessed cross-platform through a standard web browser; usability, through a responsive interface layout built with Bootstrap 5; security, through role-based session management restricting Warehouse Admin functions from Administrator-only functions; data reliability, through a normalized relational database design; and performance, so that stock quantities update immediately after an incoming or outgoing transaction is recorded.

3.3 System Design

The system design stage translated the requirement specification into Unified Modeling Language (UML) diagrams, an Entity-Relationship Diagram (ERD), and a system architecture, following the modelling practice commonly applied in information system development (Narulita *et al.*, 2024). Two main actors were defined: the Administrator and the Warehouse Admin.

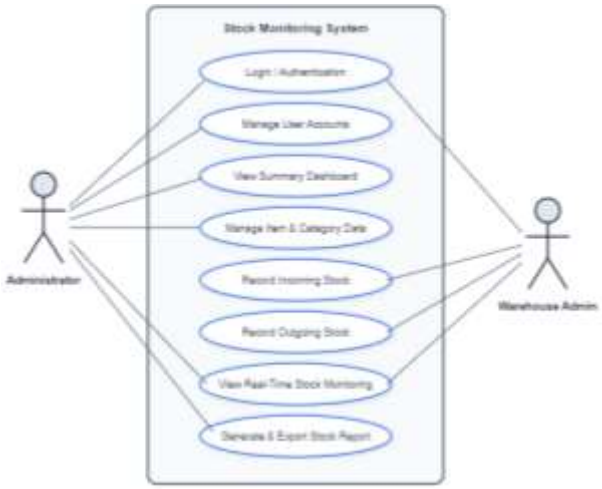

Figure 1. Use Case Diagram of the Web-Based Stock Monitoring System

Figure 1 shows the use case diagram of the system. The Administrator actor manages user accounts, views the monitoring dashboard, manages item and item-category data, reviews stock reports, and exports reports to Excel. The Warehouse Admin actor records incoming- and outgoing-stock transactions and views real-time stock monitoring, with access to the user-management and report-export functions restricted.

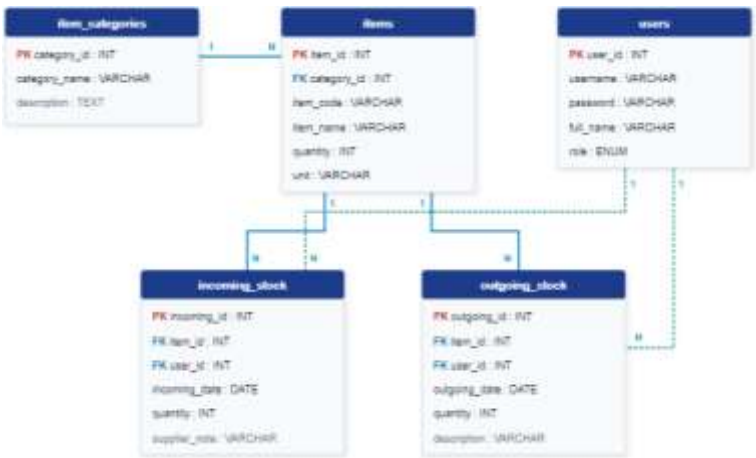

Figure 2. Entity-Relationship Diagram of the Database

Figure 2 presents the Entity-Relationship Diagram (ERD), which forms the blueprint of the relational database used by the system. The main tables are Users, Items, Item Categories, Incoming Stock, and Outgoing Stock. Referential integrity is enforced through primary and foreign keys, such as the item_id field in the Incoming Stock and Outgoing Stock tables that references the Items table, preventing redundancy and data anomalies when processing frequent stock movements.

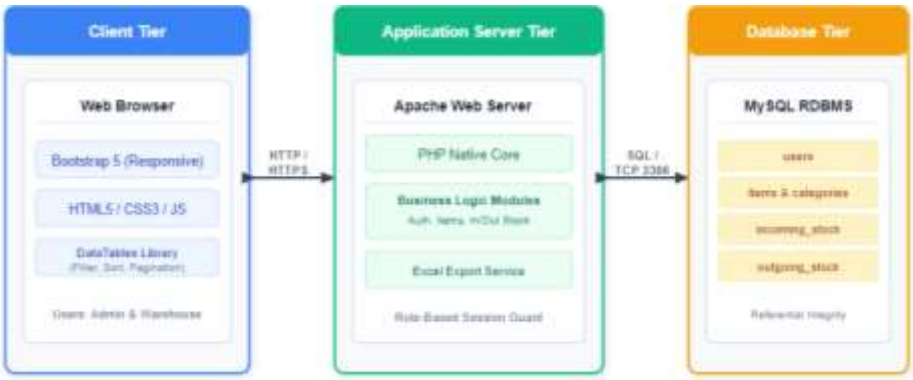

Figure 3. System Architecture of the Web-Based Stock Monitoring System

Figure 3 illustrates the client-server topology of the system. The client interface (browser) communicates over HTTP with the PHP Native application server, which processes queries to the MySQL database and renders the dashboard, item data, category data, incoming/outgoing stock records, and reports through DataTables-enabled views.

3.4 Tools and Technologies

The system was built using the technology stack summarized in Table 3, combining an open-source server-side scripting language, a relational database management system, and a responsive front-end framework.

Table 3. Hardware and Software Specification

No	Component	Technology / Specification
1	Operating System	Windows 10/11
2	Integrated Development Environment	Visual Studio Code
3	Local Web Server	XAMPP (Apache and PHP)
4	Database Management System	MySQL
5	Web Browser	Google Chrome
6	Data Table Library	DataTables (search, filter, sort, pagination)
7	Front-End Framework	Bootstrap, HTML, CSS, JavaScript
8	Development Machine	Intel Core i3/i5, 8 GB RAM
9	Reporting Library	Excel export library (e.g., PhpSpreadsheet)
10	Network	WiFi or mobile hotspot

3.5 Testing Method

System validation was carried out using two complementary techniques, namely Black Box Testing and User Acceptance Testing (UAT). Black Box Testing was used to verify the functional input-output behaviour of the system without examining the internal source code (Artino & Istiyani, 2025; Mahardika *et al.*, 2023; Shadiq *et al.*, 2021). Ten core features were tested, corresponding exactly to the functional requirements specified in Section 3.2: (1) Login, (2) User Management, (3) Item Data, (4) Item Category, (5) Incoming Stock, (6) Outgoing Stock, (7) Stock Monitoring, (8) Stock Report, (9) DataTables search, filter, sort, and pagination, and (10) Excel Export. For each feature a test scenario, an expected result, and an observed result were recorded, and a feature was declared Passed only when the observed output matched the expected output in every scenario executed. User Acceptance Testing was subsequently conducted to measure the extent to which the intended users accept the system in their actual working environment (Aliyah *et al.*, 2025; Hady *et al.*, 2020; Yakub *et al.*, 2024). The respondents were the direct users of the system at Optik Sanjaya Depok, namely the owner, who acts as the Administrator, and the warehouse staff, who act as the Warehouse Admin. The number of respondents involved was two people, consisting of the owner and warehouse staff. Data were collected through a questionnaire distributed after the respondents had operated the system directly in the daily stock-recording process for [durasi penggunaan sebelum UAT: tentukan durasi pengoperasian sistem sebelum kuesioner dibagikan, misalnya two weeks atau one month], and each questionnaire was completed independently by the respondent concerned. The questionnaire consisted of five statements covering ease of

use, clarity of the interface, suitability of the features to stock-management needs, ease of obtaining stock information, and the contribution of the system to the stock-recording process. Each statement was assessed on a five-point Likert scale, where 1 = strongly disagree, 2 = disagree, 3 = neutral, 4 = agree, and 5 = strongly agree. The score of each statement was computed as the arithmetic mean of the responses, and the mean value was interpreted using the following intervals: 1.00-1.79 (very poor), 1.80-2.59 (poor), 2.60-3.39 (fair), 3.40-4.19 (good), and 4.20-5.00 (very good).

4. Results and Discussion

4.1 Results

The web-based stock monitoring system was successfully implemented and deployed for Optik Sanjaya Depok using PHP Native, MySQL, and Bootstrap, in accordance with the design produced in Section 3.3. The implementation covers ten functional modules: login with role separation, a summary dashboard, user management, item data management, item-category management, incoming-stock recording, outgoing-stock recording, real-time stock monitoring, stock reporting with Excel export, and DataTables-based search, filter, sort, and pagination applied to every data table in the system.

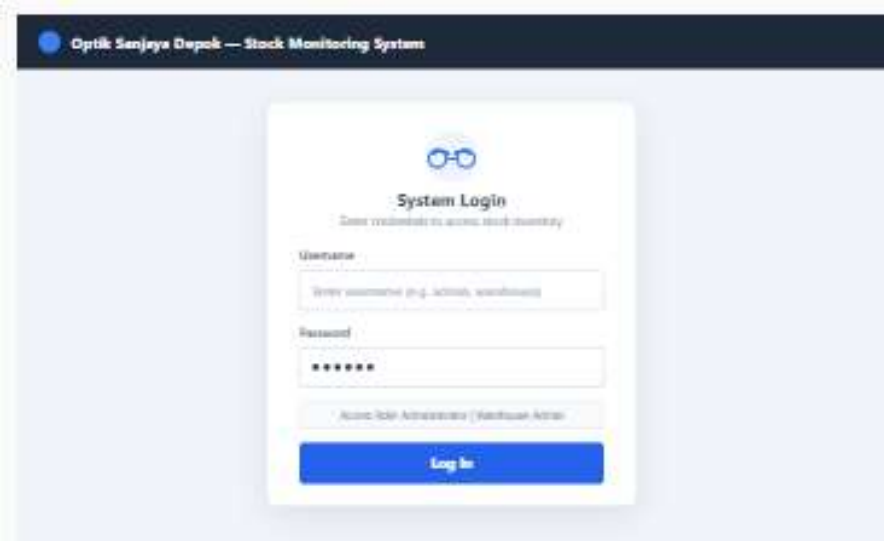

Figure 4. Login Page of the Web-Based Stock Monitoring System

Figure 4 shows the multi-user login page, which authenticates the Administrator and the Warehouse Admin separately and directs each role to the set of modules to which it is entitled. Session-based validation prevents a Warehouse Admin account from reaching the user-management and report-export functions reserved for the Administrator.

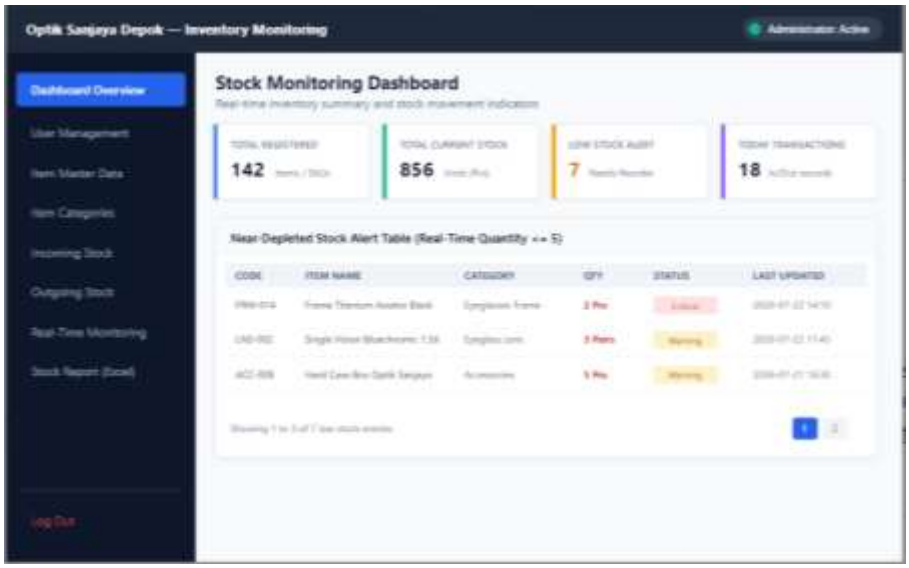

Figure 5. Monitoring Dashboard for the Administrator

Figure 5 shows the summary dashboard displayed after a successful login. The dashboard presents aggregated figures such as the total number of registered items, the total stock quantity, items whose quantity has fallen to a low level, and the most recent stock movements, so that the owner can assess the condition of the inventory without opening each module individually. The item data module stores the attributes of every pair of eyeglasses or frame, including item code, item name, item category, quantity, and unit, and provides full Create, Read, Update, and Delete operations. The item-category module manages the classification used by the item data module, so that a newly created category becomes immediately available when an item is registered. An item can be saved only when it refers to an existing category, which preserves referential integrity between the Items and Item Categories tables described in Figure 2. The incoming-stock module records the entry of goods from suppliers by referring to an item that already exists in the item data, and the quantity entered in the transaction is automatically added to the stock of that item. Conversely, the outgoing-stock module records the release of goods and automatically subtracts the recorded quantity from the stock of the corresponding item. Both modules store the transaction date and the account that performed the recording, so that every change in stock quantity can be traced back to the transaction that caused it. The stock monitoring module presents the current quantity of every item as stored in the database at the moment the page is requested, so that the displayed value always reflects the most recent incoming and outgoing transactions. The stock report module presents the recapitulation of stock and stock movement, while every data table in the system is rendered through DataTables, which provides client-side search, column sorting, filtering, and pagination without reloading the page. The Excel export function generates the displayed report as a spreadsheet file that can be opened and archived by the owner without re-typing the data. System validation through Black Box Testing was carried out on the ten core features defined in Section 3.5, and the results are summarized in Table 4.

Table 4. Black Box Testing Results

No	Feature	Test Scenario	Expected Result	Result
1	Login	Enter a registered username and password for an Administrator account and for a Warehouse Admin account	The account is authenticated and directed to the menu corresponding to its role	Passed
2	User Management	Add a new user account, assign its role, and then edit the stored account data	The account is saved with the assigned role and the change is reflected in the user list	Passed
3	Item Data	Add a new eyeglasses/frame item together with its code, name, category, and quantity	The item is stored in the database and appears in the item list	Passed
4	Item Category	Add a new item category and then select it when registering an item	The category is stored and can be selected in the item data form	Passed
5	Incoming Stock	Record an incoming-stock transaction for an item with a defined quantity	The transaction is stored and the stock of the item increases by the recorded quantity	Passed
6	Outgoing Stock	Record an outgoing-stock transaction for an item with a defined quantity	The transaction is stored and the stock of the item decreases by the recorded quantity	Passed
7	Stock Monitoring	Open the stock monitoring page after an incoming and an outgoing transaction have been recorded	The latest stock quantity of each item is displayed in accordance with the database	Passed
8	Stock Report	Open the stock report page for the recorded stock data	The report displays stock and stock-movement data consistent with the database	Passed
9	DataTables (search, filter, sort, pagination)	Enter a keyword in the search box, sort a column, and move to the next page of the item list	The data are filtered, sorted, and paginated according to the action performed, without reloading the page	Passed
10	Excel Export	Click the export button on the stock report page	The report is downloaded as an Excel file containing the same data as those displayed on the page	Passed

All ten features listed in Table 4 produced output identical to the expected result, so each of them was declared Passed. The tests confirm the two behaviours that are central to the system: the stock quantity of an item increases automatically after an incoming-stock transaction is saved and decreases automatically after an outgoing-stock transaction is saved, and the value shown on the stock monitoring page and in the stock report is consistent with the value stored in the database. No functional anomaly, such as a quantity that failed to update or a report that did not correspond to the database, was observed during the test sessions. It should be noted that black box testing verifies functional conformity with the specified requirements and does not measure performance, security, or behaviour under concurrent access, all of which were outside the scope of this study. The average score obtained for each statement is summarized in Table 5.

Table 5. User Acceptance Test Results

No	Statement	Average Score (of 5)
1	The system is easy to use	4.8
2	The interface is easy to understand	4.7
3	The features match stock-management needs	4.9
4	Stock information is easy to obtain	4.8
5	The system helps the stock-recording process	4.9

The average scores of the five statements range from 4.7 to 4.9 out of a maximum of 5, with an overall average of 4.82, which falls within the interval 4.20-5.00 and is therefore interpreted as very good. The two highest scores were obtained by the statement concerning the suitability of the features to stock-management needs and the statement concerning the contribution of the system to the stock-recording process (4.9 each), which indicates that the respondents regard the incoming-stock, outgoing-stock, and stock-monitoring modules as consistent with the way stock is actually handled in the store. The lowest score, although still within the very good category, was obtained by the statement on the clarity of the interface (4.7), which suggests that the presentation of the interface is the aspect with the greatest room for improvement.

4.2 Discussion

The results indicate that the research objectives stated in Section 1 were achieved. Objectives (1) to (6) correspond directly to the modules verified in Table 4: the system was designed and deployed, the multi-user login separated the Administrator and Warehouse Admin roles, the item and item-category modules managed the master data of the store, the incoming- and outgoing-stock modules updated item quantities automatically, the monitoring module and the dashboard presented the current stock condition, and the report module provided DataTables-based search, filter, sort, and pagination together with Excel export. The UAT scores summarized in Table 5 indicate that these modules were also accepted by the users who operate them. Compared with the related studies summarized in Table 1, this study confirms some findings and extends others. The suitability of the Waterfall model for a small-scale inventory system, reported by Ananda *et al.* (2024), Damanik *et al.* (2021), and Wau (2022), is supported by the present development process, in which the requirements of a single small store were stable enough to be specified in advance. The automatic quantity adjustment implemented in the incoming- and outgoing-stock modules corresponds to the module structure reported by Faturrachman *et al.* (2025) and Ramadhan *et al.* (2025), while the summary dashboard corresponds to the monitoring concept reported by Setiawan *et al.* (2021) and Suhada *et al.* (2026). The contribution of this study is that these two groups of features operate on the same database and under role-separated access within a single application. In contrast to the optical retail studies of Imam (2020), Indarto (2022), and Yofena (2022), which are oriented towards sales transactions, the present system addresses the internal movement of stock, so the two lines of work are complementary rather than competing. The practical contribution of the system for Optik Sanjaya Depok is that stock recording has moved from a handwritten ledger to a centralized relational database with role-based access, and that the recapitulation the owner previously had to compile manually can now be exported directly from the report module. Table 6 summarizes the operational condition of the store before and after the system was implemented.

Table 6. Comparison Before and After System Implementation

Aspect	Before Implementation	After Implementation
Stock recording	Manual, handwritten ledger	Recorded through a web form and stored in a MySQL database
Stock-data accuracy	Recording errors reported by the owner and the warehouse staff during observation and interviews; the number of errors was not documented	Quantities updated automatically by the system, so no transcription between ledger pages is required

Stock update	Delayed; the quantity changed only when the ledger was rewritten	Quantity updated automatically at the moment a transaction is saved
Data access	Single, uncontrolled access to the ledger	Role-based access for the Administrator and the Warehouse Admin
Report generation	Manual recapitulation compiled from the ledger by the owner	Report generated by the system and exported to Excel in a single step
Data search	Manual search through the pages of the ledger	DataTables search, filter, sort, and pagination

Despite these results, several limitations must be taken into account when interpreting the findings. The comparison presented in Table 6 is based on operational conditions observed during the study and reported by store personnel; the study did not collect quantitative measurements of recording time or error frequencies, meaning that improvements in stock-recording accuracy are described qualitatively rather than as percentages. Furthermore, User Acceptance Testing involved only the operational personnel of a single micro-enterprise without computing a reliability coefficient, meaning that the evaluation reflects localized user acceptance rather than statistically generalizable usability. Operationally, item identification during stock recording still requires manual selection due to the absence of integrated barcode or RFID scanning hardware. The system also remains separate from point-of-sale transactions, requiring retail sales deductions to be entered manually by warehouse staff as outgoing transactions. Additionally, automated low-stock notifications via push messaging or email are not supported, requiring the Administrator to log into the dashboard to identify near-depleted inventory. Finally, system access is restricted to desktop-oriented web browsers without native mobile application packaging, and the Excel export module is designed for administrative reporting rather than predictive stock analytics.

5. Conclusion and Recommendations

Based on the design, implementation, and testing stages carried out using the Waterfall Software Development Life Cycle (SDLC), this research successfully developed and deployed a web-based stock monitoring system for Optik Sanjaya Depok to replace its legacy handwritten ledger. The implemented system enforces multi-user access control that strictly separates the Administrator and Warehouse Admin roles, ensuring that each user accesses only authorized functional modules. The incoming- and outgoing-stock modules update inventory quantities automatically upon saving each transaction, maintaining real-time consistency across the database, the monitoring page, and the reporting views. Furthermore, the stock reporting module—equipped with DataTables search, filter, sort, and pagination alongside Excel export capabilities—provides centralized stock recapitulation without manual data re-transcription. System validation confirmed high technical and operational reliability: Black Box Testing across ten core functional features produced expected outputs in all evaluated scenarios, while User Acceptance Testing yielded statement averages between 4.7 and 4.9 and an overall average score of 4.82 out of 5.00, placing user acceptance firmly in the very good category. While these findings demonstrate that the system fulfills its functional specifications and is well accepted by operational users, this study does not provide quantitative percentages on the reduction of recording errors, as historical error frequencies were unrecorded; empirical measurement of such operational metrics remains recommended for future investigation. Building upon the operational constraints identified in Section 4.2, several directions are recommended for future development. First, integrating barcode or QR-code scanning hardware would accelerate item identification during stock-in and stock-out recording while minimizing typographical errors. Second, coupling the stock monitoring system with a dedicated point-of-sale module would allow retail customer sales to trigger automated stock deductions directly. Third, developing a mobile application or Progressive Web App (PWA) would afford warehouse personnel greater flexibility to process physical inventory movements on mobile devices. Fourth, embedding predictive analytics models could assist store management in forecasting replenishment needs from historical stock-movement patterns. Finally, incorporating automated push notifications via email or instant-messaging services would promptly alert the Administrator whenever item quantities reach critical low-stock thresholds, eliminating the need to inspect the dashboard manually.

References

Adha, N. (2022). Website inventaris barang menggunakan metode Waterfall untuk pengelolaan persediaan barang pada Bidang Prasarana dan Pengembangan Transportasi DISHUB Singkawang. *Jurnal Sistem Informasi Akuntansi*, *3*(2), 51–61. https://doi.org/10.31294/justian.v3i2.1506

Alfarizi, M., & Priyadi, W. (2024). Sistem informasi inventory kacamata berbasis web pada Toko Optik Nouglasses Bekasi. *Journal of Information Systems Management and Digital Business*, *1*(2), 176–188. https://doi.org/10.59407/jismdb.v1i2.404

Aliyah, A., Hartono, N., & Muin, A. A. (2025). Penggunaan User Acceptance Testing (UAT) pada pengujian sistem informasi pengelolaan keuangan dan inventaris barang. *Switch: Jurnal Sains dan Teknologi Informasi*, *3*(2), 42–58. https://doi.org/10.62951/switch.v3i1.330

Ananda, A., Butsianto, S., & Sulaeman, A. (2024). Penerapan metode Waterfall pada sistem informasi inventory berbasis website. *Journal of Information System Research (JOSH)*, *5*(4), 1147–1155. https://doi.org/10.47065/josh.v5i4.5669

Aprilisa, S., & Aulia, R. (2024). Penerapan metode prototype dalam pengembangan sistem informasi inventory barang berbasis web. *Jurnal Teknik Industri Terintegrasi (JUTIN)*, *7*(1), 333–340. https://doi.org/10.31004/jutin.v7i1.24749

Artino, R., & Istiyani, Y. (2025). Analisis implementasi sistem informasi pengajuan pas bandara Sultan Thaha Jambi dengan black box testing. *Bridge: Jurnal Publikasi Sistem Informasi dan Telekomunikasi*, *3*(3), 161–172. https://doi.org/10.62951/bridge.v3i3.360

Damanik, F., Meilano, R., & Wr, T. (2021). Pengembangan sistem informasi persediaan barang dengan metode Waterfall. *Jurnal Elektronika, Listrik, dan Teknologi Informasi Terapan*, *2*(2), 26–30. https://doi.org/10.37338/e.v2i2.153

Fahrizal, P., & Azhar, N. C. (2025). Perancangan sistem informasi laporan inventori bahan baku berbasis web dengan metode Waterfall pada Kedai Kopi Disiniaja. *Infomatek*, *27*(1), 153–164. https://doi.org/10.23969/infomatek.v27i1.24838

Fathoni, M., & Kustiyono. (2025). Pengembangan sistem informasi persediaan barang menggunakan model Waterfall untuk meningkatkan efisiensi pengelolaan stok. *Jurnal Algoritma*, *22*(2), 523–532. https://doi.org/10.33364/algoritma/v.22-2.2533

Faturrachman, M. S., Huda, B., Nurapriani, F., & Hilabi, S. S. (2025). Perancangan aplikasi manajemen stok barang berbasis web menggunakan metode Waterfall dengan modul barang masuk, barang keluar dan permintaan barang. *Jurnal Mahasiswa Teknik Informatika (JATI)*, *9*(5), 10189–10196. https://doi.org/10.36040/jati.v9i5.15198

Hady, E. L., Haryono, K., & Rahayu, N. W. (2020). User Acceptance Testing (UAT) pada purwarupa sistem tabungan santri (Studi kasus: Pondok Pesantren Al-Mawaddah). *Jurnal Ilmiah Multimedia dan Komunikasi*, *5*(1), 1–10. https://doi.org/10.56873/jimk.v5i1.64

Hikmal, M., Yudertha, A., & Ramadhani, Y. (2025). Perancangan sistem informasi manajemen stok material berbasis web di Gudang PLN ULP Telanaipura Jambi. *RIGGS: Journal of Artificial Intelligence and Digital Business*, *4*(3), 4845–4856. https://doi.org/10.62951/riggs.v4i3.504

Imam, B. (2020). *Sistem informasi penjualan kacamata berbasis web pada Indo Optikal Jakarta Timur* [Undergraduate thesis, Universitas Nusa Mandiri]. Repository Universitas Nusa Mandiri.

Indarto, A. K., Purwahana, R., & Syahputra, S. L. (2022). Sistem informasi penjualan kacamata toko Optik Kunanti berbasis Android. *Go Infotech: Jurnal Ilmiah STMIK AUB*, *28*(2), 195–206. https://doi.org/10.36309/goi.v28i2.179

Mahardika, F., Fitriani, A., & Al'Amin, M. (2023). Testing sistem pada Dealer Management System Service menggunakan metode black box testing. *Hello World: Jurnal Ilmu Komputer*, *2*(3), 139–144. https://doi.org/10.56211/helloworld.v2i3.364

Narulita, S., Nugroho, A., & Abdillah, M. Z. (2024). Diagram Unified Modelling Language (UML) untuk perancangan sistem informasi manajemen penelitian dan pengabdian masyarakat (SIMLITABMAS). *Bridge: Jurnal Publikasi Sistem Informasi dan Telekomunikasi*, *2*(3), 244–256. https://doi.org/10.62951/bridge.v2i3.155

Ramadhan, F. A., Amsory, M. I., Ikhsan, M., & Haryono, W. (2025). Perancangan sistem informasi manajemen stok barang berbasis web menggunakan model Waterfall pada Toko Dhanisa Mandiri Jaya. *Router: Jurnal Teknik Informatika dan Terapan*, *3*(2), 127–138. https://doi.org/10.62951/router.v3i2.509

Ridwansyah, R., Rifqie, D. M., & Nuridayanti, N. (2023). Sistem informasi inventaris toko berbasis web untuk UMKM penyewaan kostum. *Jurnal Teknologi dan Sistem Informasi Bisnis*, *5*(3), 289–295. https://doi.org/10.47233/jteksis.v5i3.874

Setiawan, A. B., Rachmawati, W., Arrahman, A. T., Natasyah, N., & Fadil, F. N. S. (2021). Aplikasi monitoring stok barang berbasis web pada PT. Intermetal Indo Mekanika. *ADI Bisnis Digital Interdisiplin Journal*, *2*(2), 1–6. https://doi.org/10.34306/abdi.v2i2.254

Shadiq, J., Safei, A., & Ratu Loly, R. W. (2021). Pengujian aplikasi peminjaman kendaraan operasional kantor menggunakan black box testing. *Information Management for Educators and Professionals*, *5*(2), 97–110. https://doi.org/10.51211/imbi.v5i2.1557

Suhada, M. R., Muchsam, Y., Firizkiansah, A., & Ardiansyah, M. (2026). Rancang bangun sistem monitoring stok barang berbasis web untuk meningkatkan efisiensi manajemen persediaan barang pada UMKM. *JIKOMTI: Jurnal Ilmiah Ilmu Komputer dan Teknologi Informasi*, *3*(1), 53–58. https://doi.org/10.62951/jikomti.v3i1.597

Susanto, D. A., & Purnomo, H. D. (2023). Perancangan sistem informasi gudang berbasis web menggunakan framework Laravel. *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, *10*(1), 63–77. https://doi.org/10.35957/jatisi.v10i1.2619

Tolampi, C. C. P., & Maria, E. (2024). Perancangan sistem informasi tracking dan monitoring posisi barang menggunakan metode Waterfall. *Jurnal Teknologi dan Sistem Informasi Bisnis*, *6*(4), 806–812. https://doi.org/10.47233/jteksis.v6i4.1616

Wau, K. (2022). Pengembangan sistem informasi persediaan gudang berbasis website dengan metode Waterfall. *Jurnal Teknik, Komputer, Agroteknologi dan Sains*, *1*(1), 10–23. https://doi.org/10.56248/marostek.v1i1.8

Yakub, H., Daniawan, B., Wijaya, A., & Damayanti, L. (2024). Sistem informasi e-commerce berbasis website dengan metode pengujian User Acceptance Testing. *JSITIK: Jurnal Sistem Informasi dan Teknologi Informasi Komputer*, *2*(2), 113–127. https://doi.org/10.53624/jsitik.v2i2.378

Yofena. (2022). Sistem informasi penjualan kacamata berbasis web pada Optik Yonotrend. *Jurnal Topik Global*, *1*(1), 1–6.