Vol. 7, No. 4, December 2025

ISI Journal of Information Systems and Informatics

Published By Asosiasi Doktor Sistem Informasi Indonesia

Utilizing ROP in a Mobile Based Warehouse Management System for Small Retail Businesses

Winni Setiawati[1], Jap Tji Beng[2], Tony[3]

[1,2,3] Department of Information System, Universitas Tarumanagara, Jakarta, Indonesia

Email: winni.825220002@stu.untar.ac.id[1], t.jap@untar.ac.id[2], tony@fti.untar.ac.id[3]

Received: Oct 11, 2025

Revised: Nov 5, 2025

Accepted: Nov 24, 2025

Published: Dec 10, 2025

Corresponding Author

Author Name*:

Jap Tji Beng, Tony

Email*:

t.jap@fti.untar.ac.id

DOI:

10.63158/journalisi.v7i4.1391

Abstract. ABC Store, a wholesale sock retailer, operates through both physical and online platforms. With increasing online sales, the company has experienced frequent stockouts due to inefficient manual inventory processes. This study aims to address these challenges by developing a mobile-based Warehouse Management System (WMS) integrated with a Reorder Point (ROP) calculation. The system was developed using the V-Shaped model, which involved requirement gathering through interviews and warehouse observations. The application was built using the lightweight Flutter framework and an SQLite local database, with testing including white-box, black-box, and User Acceptance Testing (UAT). Additionally, QR code scanning was implemented to improve document tracking and inventory management. The results indicate that the system functions as expected, with the ROP calculation effectively supporting restocking decisions and minimizing stockouts. This study contributes to the field by demonstrating the practical application of integrating simplified ROP calculation into a mobile-based WMS, highlighting its potential to improve warehouse operations for small businesses with limited infrastructure. The approach offers a scalable solution for managing inventory efficiently and cost-effectively, especially in small-scale retail settings.

Keywords: warehouse management system, reorder point, v-shaped model, qr code

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

1. INTRODUCTION

ABC Wholesale Store deals in fashion items such as socks, gloves, and many more. The store is trying to increase sales by opening an online store, but the store's internal management systems, such as stock and warehouse recording, are still done manually. This causes the store to frequently experience stockouts (running out of stock due to shortages or product backlog, which leads to the risk of lost sales) and overstocking (excess stock in the warehouse, which leads to inflated costs) [1].

One solution to overcome stockouts and overstocking is to develop and implement a warehouse management system (WMS). This system can manage inventory availability more accurately than manual methods [2]. To facilitate the use of the application in various positions within the store warehouse, the option to develop a mobile-based application is the best choice. In addition, mobile devices, including Android ones, are also very popular among Indonesians with 92 million users reported by E-Marketer in 2019 [3]. Android mobile applications have several frameworks such as Flutter that facilitate development, have high utilities and can be integrated with various features that facilitate tracking goods in warehouses, such as QR code scanning [4], [5]. Ultimately, Flutter is chosen due to its lightweight environment with low hardware requirements as most small businesses already own Android mobile devices. In Addition, SQLite provides a simple local database that enables offline data storage as online signal might be compromised inside a warehouse with limited resources. WMS can also be integrated with many features such as real-time stock keeping systems such as IoT, RFID and automated stocking to reduce the overstock and stockout risk [6], [7]. But one of the simplest ways to prevent stockout is Reorder Point (ROP) analysis that had been used in many inventory and warehouse management systems.

In previous studies, both Wairooy *et al.* [8] and Antonius *et al.* [9] used the Waterfall Software Development Life Cycle (SDLC) methodology to help develop a transaction and inventory system. [8] also documented that they utilized QR code scanning functionality in their mobile-based system and can increase their inventory recording efficiency. Other studies discussed the development of a warehouse management system and an inventory system for more sophisticated companies. Lukas *et al* [10] reported creating an inventory system that could update their information in real time between company

divisions, while Dewi *et al* [11]. explained the usage of Internet of Things (IoT) Technology inside their WMS system. Afrizal *et al* [12] Showed that they successfully integrated ROP (Reorder Point) analysis for a lightweight web application. However, most previous studies focused on developing WMS or inventory systems for large-scale enterprises or web-based environments even though both small and large business faces the same stock keeping and warehouse management challenges that are caused by disasters, bottlenecks in transporting goods or other supplier related problems yet large business will have ways to reduce such risk with dedicated team and supply chain management software [13]. Such software system often requires sophisticated infrastructure, substantial cost, which clearly are not suitable for small businesses with limited resources that typically need lightweight, mobile, and offline ready systems for easy maintenance.

Based on this gap, this study would like to know whether a lightweight mobile-based WMS integrated with a simplified ROP analysis could support stock management and operational efficiency in a small business. To address the research question, this study aims: (1) To develop a mobile-based WMS suitable for small businesses using Flutter framework and SQLite Local database. (2) To Integrate simplified Reorder Point (ROP) analysis into the system to support restocking decisions. (3) To implement QR Code scanning to support item and transaction document tracking. (4) Improve monitoring and operational mobility for small businesses with limited resources.

2. METHODS

The design of the ABC Store WMS application uses the Software Development Life Cycle (SDLC) method with a V-Shaped model, which emphasizes the relationship between the design and testing stages as illustrated in the V-Shaped diagram in Figure 1. The development stages include requirements analysis, system design, implementation, unit testing, integration testing, system testing, and user acceptance testing (UAT). The testing procedures were all done manually without using any automated tools. The application was tested on physical Android devices to evaluate user interface (UI) behavior and flow while the database was inspected with the SQLite inspector in Android Studio. While both V-Shaped and Waterfall model thrived in projects that have clear and defined requirements, the V-Shaped model was chosen since it provides clear correspondence between development phase and testing phase [14]. These correspondence benefits to

make sure all primary functions such as stock movement, ROP calculation, and QR tracking are valid. Agile methodologies, while considered, were not implemented as the project was developed solitarily within a limited timeframe and did not require continuous iteration. Software requirements were collected through an interview with one ABC store's warehouse staff who is also the current main warehouse administrator and thus has a comprehensive knowledge of the warehouse business processes. Requirement gathering was also supplemented by observing the business processes inside their warehouse. These subjective methodologies of collecting requirements are still commonly and effectively used even by other published studies such as the studies done by Fonggo *et al.* [15]. The requirements gathered were analyzed and summarized into a list of primary features that needed to be designed and implemented that can be seen on Table 1 and Table 2.

Table 1. System functional requirements

User	Functional requirements
Warehouse staff	Manage stock category
	Edit staff own profile data
	Manages item stock, stock variation, and stock location
	Manage incoming stock
	Manage outgoing stock
	User able to scan barcode to track incoming / outgoing stock
	Display stocks that need restocking (current stock < ROP) and currently stockout stock
Store Owner	Manage, see, and print report
	Edit store owner own profile data

Table 2. System non-functional requirements

Category	Non-Functional requirements
Performance	System can handle 200 types of products
Usability	User interface needs to be simple without many distracting color

Category	Non-Functional requirements
Security	Two types of users (warehouse staff and store owner) can log in to their respective account; SQLite stores the data locally with simple SHA256 encryption
Availability	Application can be use offline
Reliability	Data must not be lost after application restart
Portability	Application can run on minimal Android 11 mobile devices
Maintainability	Database must be able to be exported for backup

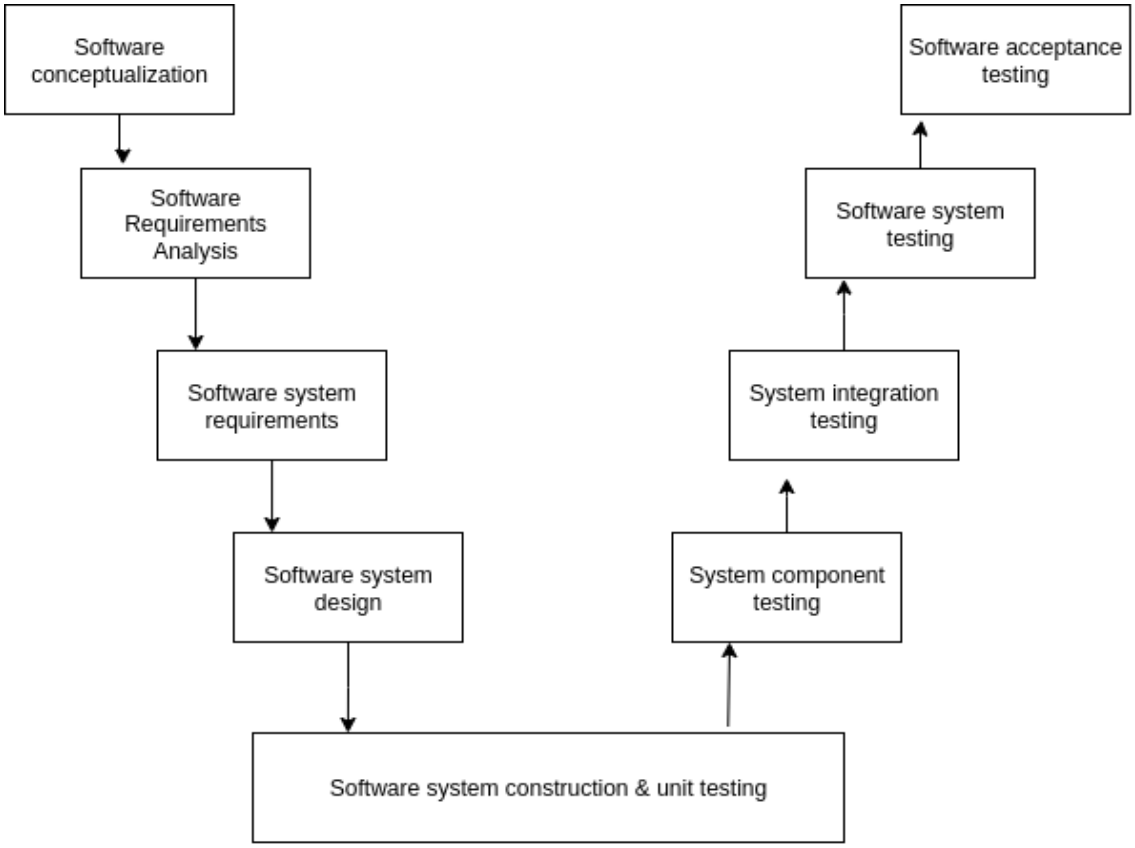

Figure 1. SDLC V-Shaped model

The Visualization of the information flow process to explain the design of the ABC Wholesale Store mobile-based WMS application uses various Unified Modeling Languages (UML), such as use case diagrams. Use case diagram is part of UML that represents the functionality of the system between the requests of actors and the use cases (functionality) of the system and have been commonly used by multiple previous development [8], [14], [16]. Within the use case diagram software applications can be used

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

by various users, namely store owners and store warehouse staff. The rest of the functionalities can be observed in Figure 2.

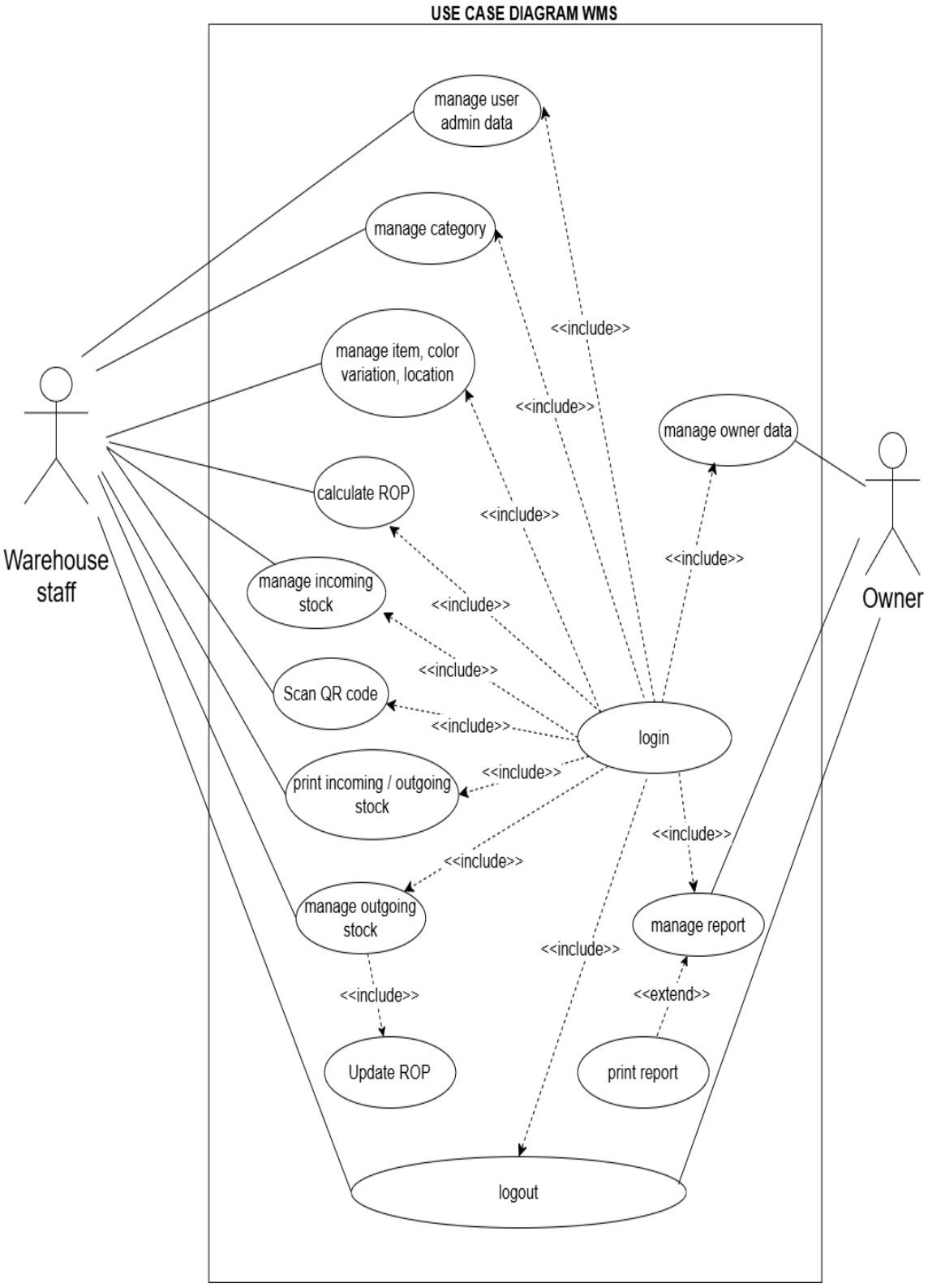

Figure 2. Use Case Diagram

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

Database design aims to create a representation of data, relationships between data, and data constraints that are appropriate for the company's conditions. It is carried out in stages, namely by designing a new conceptual database and then designing the database logically. After designing the conceptual and logical databases, a valid entity relationship diagram (ERD) without redundancy can be produced. Figure 3 presents the ERD, which illustrates the relationships between entities in the ABC Store mobile WMS application.

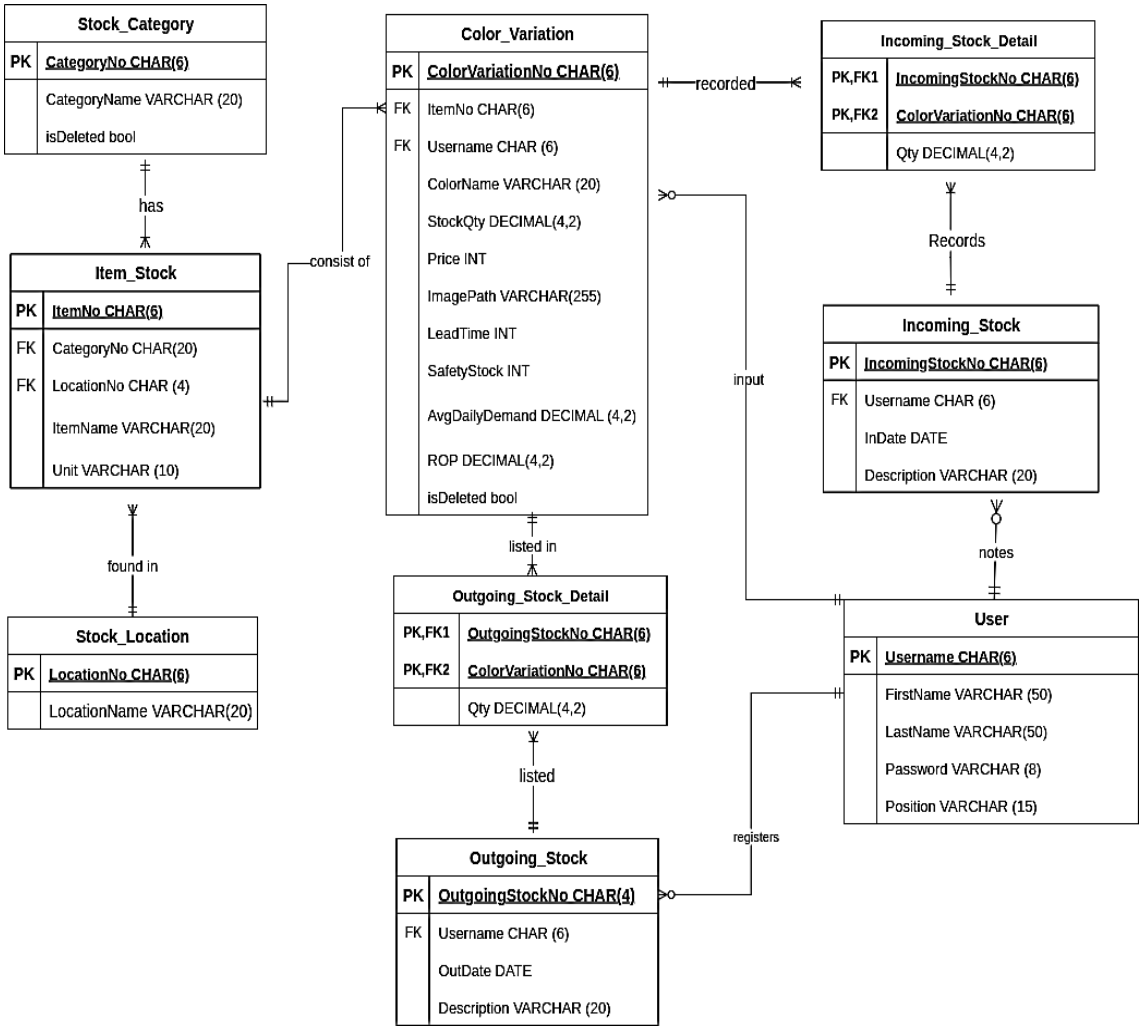

Figure 3. ERD

The application was developed using the Flutter framework and Dart programming language. Both were chosen so that developers could utilize various external package libraries, such as SQLite as a local data storage backend, mobile_scanner for the QR code scanning feature, GetX for project state management, and printing to create previews and export documents in PDF files [17]. To better illustrate the relationships of the front

end, backend and storage within the system, layered system architecture diagram has been created and can be seen in Figure 4. Once the development was finished, testing was conducted using two approaches, namely Whitebox testing and Blackbox testing. Whitebox testing directly reviewed the code structure to ensure that the logic flow, internal design, and security aspects were working properly with the help of debug console within the Integrated Development Environment (IDE). Meanwhile, black box testing focuses on testing the functionality of the system from the user's perspective without looking at the code structure [18]. Finally, UAT was conducted with the participation of end-users to ascertain whether all operational requirements had been met [19]. Similar use of UAT was also reported by Pranata *et al.* [20] to evaluate system functionality during their testing phase.

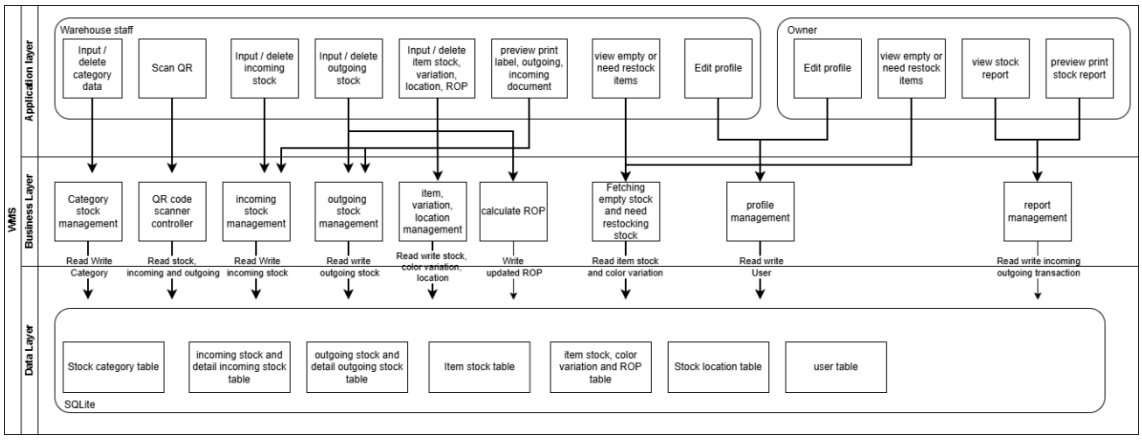

Figure 4. Layered system architecture

2.1 Reorder point

Reorder point (ROP) calculations help determine when stock should be reordered. When the stock quantity drops below that number, it indicates the need for stock replenishment. ROP values are assigned for every item color variation and calculated by multiplying the time required to receive goods (lead time) by the average daily demand and then adding the value of safety stock. For this research, the simplest method to calculate ROP was chosen to accommodate ABC store's small business scale and staff capabilities. The average daily demand is defined as the average number of outgoing stocks in the span of 30 days. Safety stock is the minimum stock that must be in the warehouse to anticipate stockouts [21]. For this reason, the safety stock value is obtained by taking the maximum stock value that has been used for 30 days. The lead time value

was obtained by asking the ABC store to estimate the duration for each item and color variation, from ordering to receiving. Equation (1) states the ROP calculation used in this research [12]:

$$\text{ROP} = \text{Average daily demand} \times \text{Lead Time} + \text{Safety Stock} \quad (1)$$

To illustrate the usage of the ROP equation, an example of the ROP calculation using the stock item cream-colored socks can be seen in Table 3. In this scenario, the item has an average outgoing demand of 2 dozen per day over a 30-day period. It is required to have at least 3 dozen socks in stock, since 3 dozen represents the maximum quantity taken from the warehouse in a single day during that period. The Item needs one day to replenish its stock. From this information and using the ROP equation above, the value of ROP for the item cream-colored socks was 5 dozen.

Table 3. Example of ROP calculation for stock of cream-colored lace socks

No	Avg daily demand	Lead time	Safety stock	ROP
1	2 dozen	1 day	3 dozen	5

3. RESULTS AND DISCUSSION

The application was developed based on the design specifications outlined in the design stage. The development process began with the implementation of the front-end interface, followed by the creation of the back-end components and database tables. Once both parts were integrated, testing was conducted to ensure that all codes, functions, and features operated according to the system requirements in accordance with the V-Shaped model. Predefined test cases were also prepared and used to maintain a structured evaluation.

3.1. User Interface

A well-designed interface for the ABC Store warehouse management system can help create successful business processes and ease of use. To plan the appearance of an application interface, a combination of simple visual components to show the layout and specific functions called a wireframe is used. Wireframe design is also used to ensure

appropriate interaction between users and system interface elements [22]. Examples of application interface displays can be seen in Figures 5 and 6.

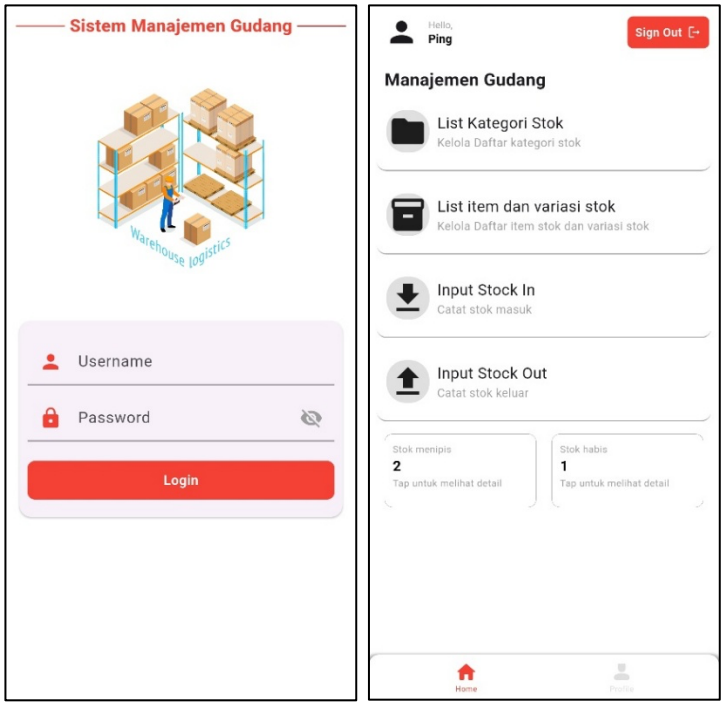

Figure 5. Login and homepage UI

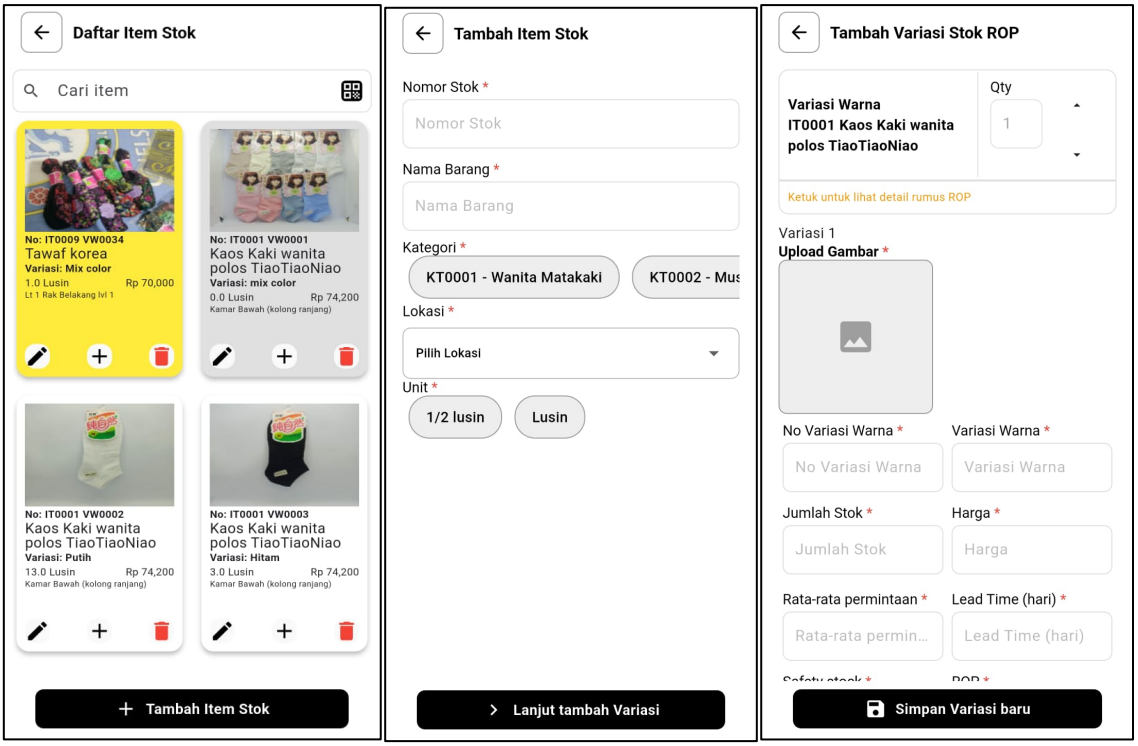

Figure 6. UI Form for managing stock data and ROP

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

These are the first pages seen and used by users to navigate within the ABC Store warehouse management application menus. Figure 5 presents the login page interface and homepage which provides navigation to the main menus for category, stock item, item variation, incoming and outgoing stock management. Additionally, the homepage also shows simple statistics indicating the number of items with low quantity and items with zero quantity. The top section of the homepage displays user's name along with a Sign out button for terminating the current session. Figure 6 shows the item and stock color variation management page where users can view their list of stocks in a grid layout, add new stock items, update existing stock, and delete unnecessary stock records. Item with low quantity is highlighted in yellow while out of stock items are marked with the color grey.

3.2. Testing

ROP calculations can be entered either manually by the warehouse staff or recalculated automatically by the system. The ROP value for each stock variation is recalculated every time an outgoing stock transaction is recorded by the system. To test this calculation, we used a stock of 5 dozen cream-colored socks, as shown in Table 4.

Table 4. Sample scenario of ROP calculation for cream-colored socks

No	Outgoing stock ID	Total outgoing stock	Current stock
Original stock: 5 dozen			
1	SK001	1 dozen	4 dozen
2	SK002	3 dozen	1 dozen

No	Outgoing stock ID	Avg daily demand	Lead time	Safety Stock	ROP
1	SK001	1 Dozen	1 day	1 dozen	2
2	SK002	2 Dozen	1 day	3 dozen	5

After the SK001 stock transaction, a new ROP value of 2 is obtained with a current stock of 4 dozen, so the current stock is still greater than the ROP value (4>2) and restocking is not yet necessary. The stock-out transaction with number SK002 results in a new ROP value of 5 and the current stock is reduced to 1 dozen, so the current stock is below the ROP value (1<5) and the system will indicate that restocking is required. The ROP recalculation test and the display showing low stock on the application are acceptable.

On implementation, Figure 7 shows that ROP calculation was running in the background each time a stock variation was in outgoing transaction with a yellow color indicator or an orange color notification.

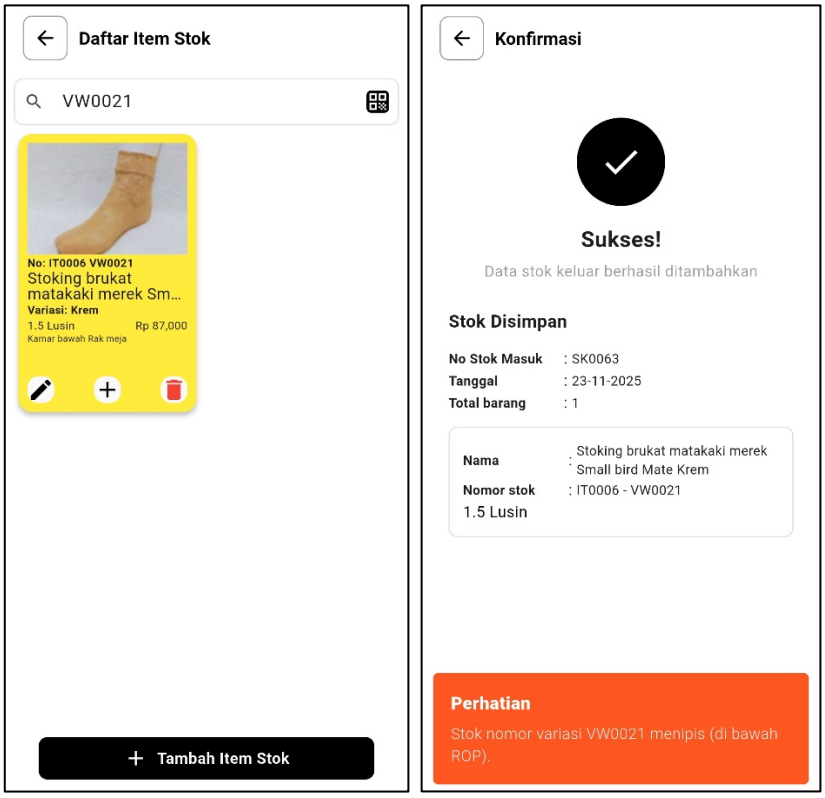

Figure 7. and an indicator of stock lower than ROP

The results of Whitebox testing by application developers for each test case were acceptable or in line with expectations, while black box testing was conducted with end users in a UAT. UAT was conducted with end users, namely warehouse administrators and store owners. Before testing, users were given user manuals on how to use the application. The UAT scenario includes testing key functions such as managing categories, managing stock variation items, managing stock reports, recording stock inflows and outflows, scanning QR codes to search for items, and calculating ROP. Based on the test results, all features can be used properly without any significant obstacles. ROP calculation was simple enough to be understood by users and successfully integrated with SQLite database and the lightweight mobile based WMS application. These results indicate that the application has met its development objectives and is ready to be implemented in the store's operational environment. The UAT result table can be seen in Table 5.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

Table 5. UAT Result

Scenario	Testing steps	Expected result	Status
Manage category	User able to see available categories, add, edit, and delete category	Stock category successfully managed	pass
Edit profile data	User is able to change their name and password	User data successfully edited	pass
Manage item stock and stock variation	User able to see stocks being displayed, search for an Item with QR Code scan, add, edit, and delete new stock	Item stock, stock variation, ROP successfully managed	pass
Manage reports	User is able to see the displayed report based on selected month	Incoming and outgoing Report are successfully displayed and printed	pass
Manage incoming stock	User is able to search incoming stock, input incoming stock, and delete unwanted incoming stock	Incoming stock successfully managed	pass
Manage outgoing stock	User is able to search outgoing stock, input outgoing stock, and delete unwanted outgoing stock	Outgoing stock successfully managed	pass
QR Code tracking	User able to scan barcode to track incoming / outgoing stock	QR code scan able to retrieve the desired transaction data	pass
Display need to restock and empty stocks	User tap the card that displayed the number of stock need to restock and empty stock	Stock need restocking and empty successfully displayed	pass

Based on the results of the testing process, every system's primary functionality was satisfactory. End users were able to utilize every module menu without many difficulties

after being provided with user manual and live demonstrations. Warehouse staff expressed positive responses toward the QR code scanning feature, particularly its ability to find items and transactions achieving average retrieval time approximately 1.392 seconds. Figure 8 shows the UI for scanning stock QR code printed on their label. The final release APK file was 35.5 MB while the SQLite database containing initial 200 product variation record occupied only 45.1 KB.

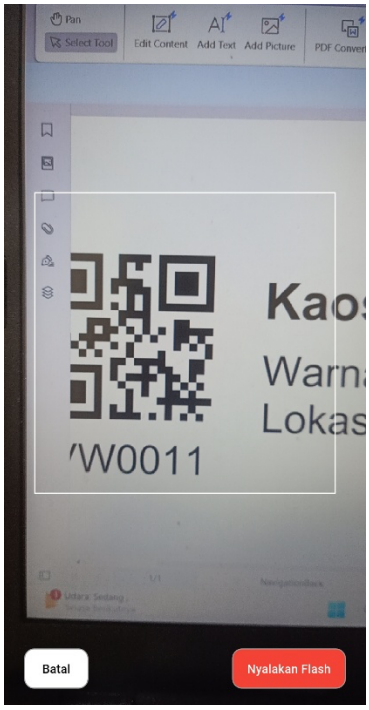

Figure 8. QR code scanning feature in retrieving items

3.3. Discussion

System provided noticeable efficiency improvements during the UAT process inside the warehouse with warehouse staff. The QR-based search significantly reduces the time needed to locate items on shelves because staff previously had to physically look for items that may take 2-5 minutes. No noticeable delays in scanning had been encounters and the flashlight features helped in brightening the QR code labels. Incoming and outgoing stock recordings also streamline information into one system instead of sporadically searching group chat notes or handwritten requests. The simplified ROP calculation has also helped contribute to the decision-making process by store owner and allows staff to recognize restocking needs way earlier than before. This finding is supported by another study conducted by Afrizal *et al*., that developed a lightweight web-based inventory system utilizing Reorder Point (ROP). Their results demonstrated that

lightweight architecture combined with ROP-based safety stock analysis can significantly enhance the competitiveness of small businesses [12]. Another way for small businesses to manage their warehouse operations with minimal resources is by outsourcing the system to a third-party provider such as Odoo. Prakoso *et al.* have shown that adopting such solutions can lead to noticeable improvements in inventory accuracy, reductions in stock outs and overstock, better decision-making, and increased operational efficiency, while also offering scalability and flexibility [23]. Overall, these improvements suggest that this system contributes to operational small-scale warehouse environments even without quantitative time measurement testing within ABC store.

Several challenges were encountered during development. First, SQLite's limited data type support, especially the absence of date data type, required custom formatting whenever app function needs to fetch data based on duration or date such as getting outgoing detail data to recalculate ROP or filtering reports based on month. Second, the positive reaction toward QR scanning feature might indicate positive prospect towards improving work efficiency inside the warehouse though no additional fraud-resistant authentication was implemented inside the code, such as neural network-based authentication or digital watermarking, and it might proceed with some security problem if warehouse staff tried to scan unknown QR code using the WMS app [24]. Finally, the whole development was done in a short time frame of 6 weeks with limited manpower which constrained the refinement of UI or features yet the core functions were successfully delivered.

The results demonstrate that the system's core functionalities operate effectively during each phase of testing; however, several points of contentions can be highlighted for discussion regarding real-world warehouse usage and system adaptability. The warehouse staff might have been able to utilize the mobile application after a brief onboarding process that might indicate that the whole system might not be fully intuitive in nature. Meanwhile, the ROP calculation was shown to successfully trigger stock warning and color indicator after each outgoing transaction though limitations due to manually input lead time might prose a possibility of inconsistent values. To correct such possibilities, a feature to edit lead time, average daily demand, and safety stock were added. Another limitation that could be highlighted is the usage of local SQLite database. Even though the usage of SQLite shows a very light storage footprint with only 45.1 KB

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

inside the final 35.5 MB release APK file, it could only provide offline database storage and no real-time multi-user synchronization functionality as they are serverless and embedded directly inside the host application [25], [26]. ABC store had no issue in this matter since they do not require those features inside their warehouse due to their small scale and mostly offline operation. But it might not be realistic for other stores or warehouses.

4. CONCLUSION

Store mobile-based warehouse management system has been successfully designed, implemented, and tested according to the specifications outlined during the planning phase. The application has demonstrated its ability to meet the requirements, functioning as intended to support efficient warehouse management. By incorporating key features such as the simplified Reorder Point (ROP) calculation, the system effectively aids in maintaining optimal stock levels and preventing shortages. Additionally, it provides a valuable tool for making more informed decisions regarding warehouse operations. The integration of QR code scanning for item tracking and transaction management further enhances the system's capability to streamline warehouse processes, improving overall operational efficiency. The main contributions of this study are the integration of the simplified ROP calculation directly into the mobile WMS, the incorporation of QR code scanning for improved tracking, and the choice of a lightweight mobile-based architecture using Flutter and SQLite. These features, in combination, make the application particularly well-suited for small businesses like ABC Store, where efficient and cost-effective solutions are essential for managing warehouse operations.

Looking ahead, there are several opportunities for future improvements to further enhance the functionality of the system. Key recommendations include adding supplier management features to facilitate better coordination with vendors, enhancing the analytics features by incorporating demand forecasting to improve inventory management, and integrating a cloud-based or hybrid database to enable real-time synchronization and multi-user capabilities. These improvements would not only enhance the user experience but also provide more advanced tools to support growth and scalability for businesses like ABC Store.

ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to Tarumanagara University for providing the opportunity and support throughout this study. We also extend our appreciation to Ms. Tasya Mulia for her valuable feedback and guidance during the preparation of this paper.

REFERENCES

[1] Z. U. Rizqi and A. Khairunisa, "Integration of Deterministic and Probabilistic Inventory Methods to Optimize the Balance Between Overstock and Stockout," in IOP Conference Series: Materials Science and Engineering, Institute of Physics Publishing, Jan. 2020, pp. 1–6. doi: 10.1088/1757-899X/722/1/012060.

[2] D. Minashkina and A. Happonen, "Warehouse Management Systems for Social and Environmental Sustainability: A Systematic Literature Review and Bibliometric Analysis," Logistics, vol. 7, no. 3, Sep. 2023, doi: 10.3390/logistics7030040.

[3] E. Damayanti, A. Asfar, F. Jama, and Suhermi, "Penggunaan Smartphone dan Perilaku Sosial pada Remaja," Window of Nursing Journal, vol. 5, no. 1, pp. 48–54, 2024, doi: 10.33096/won.v5i1.698.

[4] F. R. Ferdiansyah and R. Sofian, "Implementasi Teknologi QR Code dan Location Based Service pada Presensi Mobile Flutter," Jurnal Ilmiah Intech : Information Technology Journal of UMUS, vol. 5, no. 2, pp. 85–97, 2023, doi: 10.46772/intech.v5i2.1032.

[5] M. van Geest, B. Tekinerdogan, and C. Catal, "Design of a Reference Architecture for Developing Smart Warehouses in Industry 4.0," Comput Ind, vol. 124, Jan. 2021, doi: 10.1016/j.compind.2020.103343.

[6] A. Immadisetty, "Real-Time Inventory Management: Reducing Stockouts and Overstocks in Retail," Journal of Recent Trends in Computer Science and Engineering (JRTCSE), vol. 13, no. 1, pp. 77–88, Feb. 2025, doi: 10.70589/JRTCSE.2025.13.1.10.

[7] A. Farooq, A. B. N. Abbey, and E. C. Onukwulu, "Inventory Optimization and Sustainability in Retail: A Conceptual Approach to Data-Driven Resource Management," International Journal of Multidisciplinary Research and Growth Evaluation, vol. 5, no. 6, pp. 1356–1363, 2024, doi: 10.54660/.ijmrge.2024.5.6.1356-1363.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

[8] I. K. Wairooy, I. Dillwyn, K. P. Yonathan, and A. Lay, "Development of Mobile QR Warehouse Management Apllication Based on Flutter and Firebase," Engineering, MAthematics and Computer Science Journal (EMACS), vol. 6, no. 1, pp. 39–44, Jan. 2024, doi: 10.21512/emacsjournal.v6i1.10921.

[9] W. Antonius, D. Arisandi, and J. P. Novario, "Pembuatan Sistem Pencatatan Transaksi dan Inventori Berbasis Web Pada Toko Besi Irdawanti," Jurnal Ilmu Komputer dan Sistem Informasi, vol. 13, no. 1, pp. 1–6, 2025, doi: 10.24912/jiksi.v13i1.32880.

[10] Lukas, R. A. Praptiwi, A. T. Priandika, Suaidah, and D. Alita, "Implementasi REST API pada Manajemen Stok Barang Berbasis Aplikasi Web (Studi Kasus: PT Jon Kuliner Indonesia)," Jurnal Teknik Komputer, vol. 3, no. 1, pp. 19–24, 2024, doi: 10.14710/jtk.v3i1.46234.

[11] I. K. Dewi and R. Nur Shofa, "Development of Warehouse Management System to Manage Warehouse Operations," Journal of Applied Information Systems and Informatics, vol. 1, no. 1, pp. 15–23, Nov. 2023, doi: 10.37058/jaisi.v1i1.8991.

[12] N. Afrizal, J. Minardi, and D. Mahendra, "Safety Stock and Reorder Point System for RF Media Stock Optimization," Scientific Journal of Informatics, vol. 12, no. 2, pp. 339–348, Jul. 2025, doi: 10.15294/sji.v12i2.22232.

[13] N. Natesh Kumar, "The Supply Chain Maze: The Struggles of Small and Medium Enterprises and Ways to Navigate the Challenges," International Journal of Supply Chain Management, vol. 14, no. 2, pp. 33–45, Apr. 2025, doi: 10.59160/ijscm.v14i2.6304.

[14] M. O. Tumbade, K. D. Hartomo, and H. D. Purnomo, "Pengembangan Perangkat Lunak Berbasis Website Menggunakan Kombinasi Metode Scrum Dan V-Model," Jurnal Teknologi Informasi dan Ilmu Komputer, vol. 11, no. 3, pp. 447–460, Jul. 2024, doi: 10.25126/jtiik.938456.

[15] F. Fonggo, J. T. Beng, and D. Arisandi, "Web-based canteen payment and ordering system," in IOP Conference Series: Materials Science and Engineering, IOP Publishing Ltd, Dec. 2020. doi: 10.1088/1757-899X/1007/1/012159.

[16] R. Rifnaldy and Tony, "Perancangan Aplikasi Media Informasi dan Pemesanan Berbasis Web untuk Coffee Shop Tempat Bercerita," Jurnal Ilmu Komputer dan Sistem Informasi, vol. 11, no. 1, pp. 1–7, 2023, doi: 10.24912/jiksi.v11i1.24141.

[17] A. Tewari and P. Singh, "Android App Development: A Review," Journal of Management and Service Science, vol. 01, no. 006, pp. 1–6, 2021, doi: 10.54060/JMSS/001.02.006.

[18] D. Wintana, D. Pribadi, and M. Y. Nurhadi, "Analisis Perbandingan Efektifitas White-Box Testing dan Black-Box Testing," Jurnal Larik, vol. 2, no. 1, pp. 8–16, 2022, doi: 10.31294/larik.v2i1.1382.

[19] N. A. Vanesha, R. Rizky, and A. Purwanto, "Comparison Between Usability and User Acceptance Testing on Educational Game Assessment," Jurnal Sisfokom (Sistem Informasi dan Komputer), vol. 13, no. 2, pp. 210–215, Jun. 2024, doi: 10.32736/sisfokom.v13i2.2099.

[20] E. B. Pranata and T. Tony, "Utilizing ORB Algorithm in Web-Based Sales Application," Journal of Information Systems and Informatics, vol. 6, no. 1, pp. 378–398, Mar. 2024, doi: 10.51519/journalisi.v6i1.671.

[21] P. N. Anggraini, K. Harianto, and H. Sutapa, "Implementation of Safety Stock in Cracker Inventory Control at PT Merpati Mas Nusantara 2022 - 2025 in Kabupaten Kediri," Musytari, vol. 19, no. 10, pp. 1–11, 2025, doi: 10.8734/musytari.v1i2.365.

[22] A. Almani and O. Alrwais, "The Role of Wireframes in Enhancing User Interface Design," International Research Journal of Innovations in Engineering and Technology, vol. 08, no. 12, pp. 134–140, 2024, doi: 10.47001/irjiet/2024.812020.

[23] B. S. Prakoso, A. F. N. Masruriyah, R. T. Prasetyo, Mardiah, S. B. Ardika, and Feriadi, "Optimizing a Warehouse Management System (WMS) for Small and Medium Enterprises (SMEs) using Odoo ERP," in International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS), IEEE, 2024, pp. 880–886. doi: 10.1109/ICIMCIS63449.2024.10957438.

[24] S. A. Alsuhibany, "Innovative QR Code System for Tamper-Proof Generation and Fraud-Resistant Verification," Sensors, vol. 25, no. 13, Jul. 2025, doi: 10.3390/s25133855.

[25] K. P. Gaffney, D. R. Hipp, M. Prammer, D. Kennedy, L. Brasfield, and J. M. Patel, "SQLite: Past, Present, and Future," in Proceedings of the VLDB Endowment, VLDB Endowment, 2022, pp. 3535–3547. doi: 10.14778/3554821.3554842.

[26] D. Kengalagutti and Chethana, "Comparing Database Management Systems: MySQL, PostgreSQL, SQLite," International Research Journal of Engineering and Technology, 2020.