Comprehensive Journal of Science
Volume (11), Issue (42), (Sapt, 2026)
ISSN: 3104-6266

مجلة العلوم الشاملة
المجلد (11) العدد 42 سبتمبر 2026
ردمد: 3104-6266

Design and Evaluation of an Integrated Electronic System for Prescription Management, User–Pharmacy Connectivity, and Drug Inventory Management in Libya

Huda Dakhl Allah Jomma[1]* and Suheer Ramadhan Abdulsalam[1]
[1] Department of Computer, College of Electronic Technology, Tripoli, Libya
Huda Dakhl Allah Jomma, Ph.D.; Lecturer
Suheer Ramadhan Abdulsalam: Bachelor's degree in Software Engineering; Graduate Engineer
*Corresponding author: huda73jomma@gmail.com

Received: 12/9/2026 Revised: 15/9/2026 Accepted: 31/10/2026 Publication: 14/11/2026

ABSTRACT

Reliable medicine searches depend on linking availability information to the stock committed to an order. Roshita was developed and evaluated in Libya to help users find nearby pharmacies with the required medicines and connect their orders to pharmacy inventory management. Requirements came from 150 questionnaire responses and interviews and observation in five Tripoli pharmacies. Development followed an Agile incremental approach. The system comprises three applications sharing Firebase services: an Android mobile application for users, a Windows desktop application for pharmacy management, and a React web application for administration. Backend transactions coordinate reservations, partial offers, invoices, collection, and internal insurance records. Evaluation covered functionality, integration, transaction integrity, access control, and performance. Functional and component integration tests passed in 26/26 user cases, 39/40 pharmacy cases, and 15/15 administrative cases. All 12 executed system integration scenarios passed out of 14 planned. Targeted tests confirmed single allocation of contested stock, prevention of duplicate invoices, and consistent terminal order outcomes; three pharmacy corrections passed retesting. A five-minute local-emulator workload reaching 25 virtual users recorded an order-confirmation P95 of 331.29 ms against a 3,000-ms limit, with a maximum of 10.94 seconds. The error rate was 1.10%, exceeding the 1% ceiling. Dashboard loading averaged 3,287.46 ms against a 3,000-ms target. These findings support functional feasibility and the tested transaction controls. The contribution is an evaluated workflow connecting pharmacy selection by medicine availability and distance with stock reservation and fulfilment across three applications, providing a foundation for field evaluation.

KEYWORDS: drug inventory management, medicine availability, stock reservation, transaction integrity.

الملخص:

تعتمد موثوقية البحث عن الأدوية على ربط معلومات توافرها بالكميات المخصصة للطلبات. طُوّر نظام «روشيتة» وقُيّم في ليبيا لمساعدة المستخدمين على العثور على صيدليات قريبة تتوافر لديها الأدوية المطلوبة، وربط طلباتهم بإدارة مخزون الصيدليات. استُخلصت المتطلبات من 150 استجابة لاستبيان، ومن مقابلات وملاحظة في خمس صيدليات بمدينة طرابلس. واتُّبع في التطوير نهج رشيق تزايدي. ويتكون النظام من ثلاثة تطبيقات تشترك في خدمات Firebase: تطبيق للهواتف بنظام Android للمستخدمين، وتطبيق مكتبي بنظام Windows لإدارة الصيدليات، وتطبيق ويب للإدارة باستخدام React. وتنسق المعاملات الخلفية عمليات حجز الكميات والعروض الجزئية والفواتير والتحصيل وسجلات التأمين الداخلية. شمل التقييم الوظائف والتكامل وسلامة المعاملات والتحكم في الوصول والأداء. نجحت اختبارات الوظائف وتكامل المكونات في 26 من أصل 26 حالة لتطبيق المستخدم، و39 من أصل 40 حالة لتطبيق الصيدلية، و15 من أصل 15 حالة للتطبيق الإداري. ونجحت جميع سيناريوهات تكامل النظام الاثني عشر المنفذة من أصل 14 سيناريو مخططا. وأكدت اختبارات مخصصة حجز الكمية المتنازع عليها لطلب واحد فقط، ومنع تكرار الفواتير، واتساق النتائج النهائية للطلبات؛ كما نجحت إعادة اختبار ثلاثة تصحيحات في تطبيق الصيدلية. وخلال اختبار حمل مدته خمس دقائق باستخدام محاكيات محلية، وصل إلى 25 مستخدما افتراضيا، بلغ زمن الاستجابة وفق مؤشر (P95) لتأكيد الطلب 331.29 مللي ثانية، مقابل حد قدره 3,000 مللي ثانية، وبلغ الحد الأقصى 10.94 ثانية. وبلغ معدل الأخطاء 1.10% متجاوزا السقف المحدد البالغ 1%. أما متوسط تحميل لوحة المعلومات فبلغ 3,287.46 مللي ثانية مقابل هدف قدره 3,000 مللي ثانية. وتدعم هذه النتائج الجدوى الوظيفية وضوابط المعاملات المختبرة. وتتمثل مساهمة الدراسة في مسار عمل خضع للتقييم، يربط اختيار الصيدلية بحسب توافر الأدوية والمسافة بحجز المخزون وتلبية الطلبات عبر التطبيقات الثلاثة، بما يوفر أساسا للتقييم الميداني.

الكلمات المفتاحية: إدارة المخزون الدوائي؛ توافر الأدوية؛ حجز المخزون؛ سلامة المعاملات.

1 INTRODUCTION

People who need several prescribed medicines must compare availability, cost, and travel distance. Search results can become unreliable if the last available quantity is sold before confirmation. An effective service should connect the process of finding medicines with the transactions that update pharmacy stock.

Digital health guidance addresses feasibility, acceptability, resources, equity, and potential harms (World Health Organization, 2019). Pharmacy-location research links availability to distance, cost, and opening hours (Wondmagegn et al., 2023), while stock-reporting interventions show the value of inventory visibility (Barrington et al., 2010). Prescription text recognition can also reduce data-entry effort (Soeno et al., 2024).

The local requirements investigation identified a related need. Among 150 respondents, 88.0% reported frequent or occasional difficulty finding all required medicines, and 73.3% identified their unavailability in one pharmacy as a problem. Interviews and observation in five Tripoli pharmacies connected these concerns to inventory work, staff permissions, and order handling.

1.1 Research Problem and Questions

The research examines the transition from an availability enquiry to a traceable order based on current stock. Concurrent requests, outdated partial offers, repeated submissions, and cancellations after invoicing require coordinated inventory and financial rules. Two questions guide the study:

RQ1: How can prescription-assisted medicine discovery be combined with stock reservation and pharmacy fulfilment while maintaining consistent quantities, authorised state transitions, and traceable financial records?

RQ2: To what extent does the implemented system satisfy its functional, integration, transaction-integrity, access-control, and performance requirements under the tested conditions?

1.2 Objectives and Contribution

The objectives are to connect reviewed prescription input and manual search with pharmacy comparison and fulfilment; implement consistent stock transitions, repeat-safe commands, role-based access, and traceable accounting; and evaluate these controls through requirements-linked functional, concurrency, failure, and performance tests.

Roshita contributes an evaluated workflow that helps users find nearby pharmacies with the medicines they need and links their orders to pharmacy inventory management. The user mobile application, pharmacy desktop application, and administrative web application coordinate these activities through shared backend services. Its rules distinguish discovery from allocation, preparation from collection, and overdue alerts from invoice reversal. The model incorporates inventory movements, staff authority, shifts, and returns, with observable outcomes and recorded corrections supporting its evaluation.

2 CONCEPTUAL FRAMEWORK AND RELATED LITERATURE

2.1 Medicine Discovery and Inventory Visibility

Wondmagegn et al. developed a pharmacy-location framework combining medicine availability, distance, medicine cost, travel time and cost, and opening hours. They evaluated it with simulated patients and pharmacies (Wondmagegn et al., 2023). Their work informs Roshita's discovery stage; this study examines the allocation and fulfilment stages that follow selection.

Barrington et al. evaluated SMS for Life in 129 Tanzanian health facilities over 21 weeks (Barrington et al., 2010). Weekly stock reports, electronic mapping, and management follow-up improved inventory visibility. The proportion of facilities lacking at least one monitored anti-malarial medicine fell from 78% to 26%. This field evidence highlights the value of dependable stock information, without serving as a direct performance benchmark for Roshita.

2.2 Prescription Input and Pharmacy Workflow

In a simulation with 38 healthcare professionals, Soeno et al. observed median recording times of 18 seconds with OCR and 144 seconds with manual entry for six medicines; character errors were evaluated separately (Soeno et al., 2024). Roshita likewise uses extraction to assist data entry, followed by review of medicine names.

Lundhaug et al. found that a Norwegian electronic prescription system simplified some tasks while increasing coordination work for other professionals (Lundhaug et al., 2025). This supports evaluating connected workflows, including pharmacy checking and data maintenance, rather than user effort alone.

Sattar et al. assessed 23 Pakistani pharmacy applications using the Mobile App Rating Scale, covering engagement, functionality, aesthetics, and information quality (Sattar et al., 2025). These dimensions complement transaction testing. Table 1 compares the scope and evidence of these studies rather than ranking their performance.

Table 1: Position of Roshita in relation to selected studies.

Study	Main evidence	Relationship to this study
Wondmagegn et al. (2023)	Medicine-location framework tested with simulated patients and pharmacies.	Informs discovery; Roshita evaluates subsequent reservation and fulfilment.
Barrington et al. (2010)	Stock reporting in 129 facilities during a 21-week field pilot.	Shows the value of inventory visibility; the present evaluation concerns software behaviour.
Soeno et al. (2024)	OCR and manual-entry comparison with time and character-error measures.	Provides a model for evaluating reviewed prescription input.
Lundhaug et al. (2025)	Observation and interviews on cross-professional prescription workflows.	Motivates evaluation of work distributed across users and pharmacy staff.
Sattar et al. (2025)	Structured quality assessment of 23 pharmacy applications.	Complements transaction tests with usability and information-quality assessment.

2.3 Research Gap and Design Position

The reviewed studies address medicine discovery, stock visibility, prescription input, professional workflow, and application quality. Roshita connects these areas through explicit rules for allocation, invoicing, collection, and reversal. Its contribution lies in implementing and evaluating these rules within the local pharmacy workflow, including overlapping and repeated actions.

3 MATERIALS AND METHODS

3.1 Research Design and Development Process

The study draws on selected stages of the design-science methodology described by Peffers et al., focusing on problem identification, objectives, construction, demonstration, and evaluation (Peffers et al., 2007). Software development used an Agile incremental process. The user, pharmacy, and administrative applications were developed successively, followed by system integration and review.

The evidence includes requirements findings, implementation descriptions, links between requirements and tests, execution records, and documented corrections and retests. Final counts incorporate successful retests only for identified cases. Pharmacy and administrative latency statistics were checked against k6 output and browser timing records; no additional experiments were introduced.

3.2 Requirements Elicitation

A Google Forms questionnaire gathered 150 responses through convenience recruitment on social media between 25 August and 6 September 2025. It covered search difficulties, information and input preferences, ordering, communication, and reminders. Respondents aged 18–25 accounted for 73.3% of the sample. Frequencies and percentages informed the requirements in Table 2.

In December 2025, interviews and observation in five pharmacies in Al-Hadba and Hay Dimashq, Tripoli, explored stock records, enquiries, permissions, and adoption concerns with pharmacists and inventory staff. The findings refined operational workflows and clarified responsibility for updating availability information.

Table 2: Questionnaire findings and resulting design decisions (n = 150).

Reported finding	Percentage	Requirements implication
Frequent or occasional difficulty finding all medicines	88.0%	Search availability across participating pharmacies.
Unable to find all medicines in one pharmacy	73.3%	Make complete and partial availability explicit.
Difficulty knowing which pharmacy has an item	60.7%	Link discovery to pharmacy stock information.
Price requested in search results	82.0%	Display estimated prices for comparison.
Complete-list availability requested in results	74.0%	Support filtering for all requested medicines.
Distance requested in search results	65.3%	Provide location-based comparison.

Several questions allowed respondents to select more than one answer. The percentages describe the convenience sample and are not population estimates.

The questionnaire stated an academic purpose and requested no sensitive personal information. Software evaluation used test accounts and records. The study reports software outcomes, with no patient-level clinical evaluation.

3.3 Evaluation Strategy and Measures

Requirements-linked black-box tests covered boundary values, state transitions, access, concurrency, and performance. ISO/IEC 25010 and ISO/IEC/IEEE 29119-1 provide the reporting terminology (International Organization for Standardization & International Electrotechnical Commission, 2023; International Organization for Standardization et al., 2022); the study makes no certification claim. Functional cases that exercise component integration are counted once, with system integration reported separately.

RQ1 is examined through the architecture, order model, and integrity tests. For RQ2, results are compared with specified expectations and thresholds, distinguishing passed, failed, partly exercised, and unexecuted cases.

Four HTTP commands were tested separately with k6 against local Firebase emulators (Grafana Labs, n.d.). Each five-minute workload reached 25 virtual users. The detailed specification lists 30 seconds to reach 10 users, a 60-second hold, 30 seconds to reach 25, and a 180-second hold. Acceptance required P95 ≤ 3,000 ms and unexpected errors ≤ 1%. Administrative interface tests used ten repetitions and a 3,000-ms target for dashboards and searches, and five repetitions and a 10,000-ms target for maps and reports.

4 SYSTEM DESIGN AND IMPLEMENTATION

4.1 Architecture and Responsibilities

Three applications share backend services (Figure 1). The Android mobile application uses Flutter and Dart for medicine input, pharmacy comparison, orders, communication, and local reminders. The Windows desktop application uses the same framework to manage stock, purchases, sales, employees, shifts, invoices, and orders. The React web application, developed with Vite, provides administrative approval, oversight, and reporting.

The mobile application supports account and profile management, optional health and medication records, family beneficiaries, and their insurance details. Users can enter medicines manually or review names extracted from a prescription, archive and retrieve prescriptions, and compare or filter pharmacies by availability, distance, estimated price, opening status, and accepted insurance. Similar-name results require user review. Users select a beneficiary when placing an order, review its items and quantities, respond to partial offers, track preparation and collection, and cancel before invoicing. Order-linked messages and notifications support communication with the pharmacy. Received medicines can be added to the beneficiary's medication history. Local medicine and appointment reminders can be created, edited, paused, resumed, or removed.

Within this architecture, the desktop application provides an integrated system for day-to-day pharmacy management. It supports pharmacy registration and settings, staff accounts and permissions, salary records and payments, and medicine records with prices, sale units, batches, expiry dates, barcode search, and data import and export. Supplier records, purchase invoices, and supplier balances are linked to inventory updates. Authorised stock adjustments record corrections, damage, and expiry.

Point-of-sale transactions, receipts, full or partial returns, and shift opening, reconciliation, and closure are handled alongside mobile orders and their associated messages. Insurance-company records, coverage rates, receivables, and collections are maintained with expenses and financial indicators. Dashboards, operational and administrative notifications, audit records, and printable or exportable reports support oversight. Stock movements and active reservations feed the availability information used by the mobile application.

The administrative web application supports the review of pharmacy registration documents and decisions to approve, reject, or request completion. Administrators monitor pharmacy status and connectivity, suspend or restore pharmacy visibility in searches, and review user accounts and order histories within defined access limits. The dashboard distinguishes pharmacy processing delays from late customer collection. A coverage map shows pharmacy locations, while aggregate reports describe orders, pharmacy activity, user growth, and frequently requested medicines. Administrators can send notifications to users, pharmacies, or both, monitor sending outcomes, and review audit records of administrative decisions. These functions provide oversight without exposing private health records or allowing administrators to perform pharmacy order operations.

Firebase Authentication verifies user identity, Firestore stores operational documents, and Cloud Storage holds files and attachments. Before permitting sensitive commands, backend functions check account status, session authority, pharmacy membership, permissions, and business state. Firestore and Storage rules restrict client operations. Privileged server libraries bypass client Security Rules, so server-side validation is required (Google, n.d.-d).

Derived indexes support discovery, while stock and reservation records govern allocation. Scheduled functions perform time-based checks. Notifications and index updates follow committed changes, so notification delivery delays do not alter order outcomes. Repeated or out-of-order Firestore events must be considered at this boundary (Google, n.d.-a).

4.2 Prescription Input and Medicine Discovery

Google ML Kit extracts text from supported prescription images on the device, while Syncfusion handles text-bearing PDFs. Extracted text is sent over HTTPS to a Flask service for rule-based cleaning and matching against a CSV dictionary of medicine names. Users review and correct the names before searching. Uploading a prescription attachment is a separate operation.

Manual entry remains available. ML Kit supports Latin and several Asian scripts, but not Arabic (Google, n.d.-b). If Flask is unavailable, manually confirmed names can still query the shared search index when cloud access is available. Clinical interpretation and medicine-substitution decisions remain the responsibility of qualified professionals.

Search results combine availability, estimated prices, distance, and recorded insurance acceptance, using OpenStreetMap-based components and location services. Searching does not reserve stock. Order submission rechecks pharmacy eligibility, including visibility and acceptance of new orders. Confirmation rechecks current quantities.

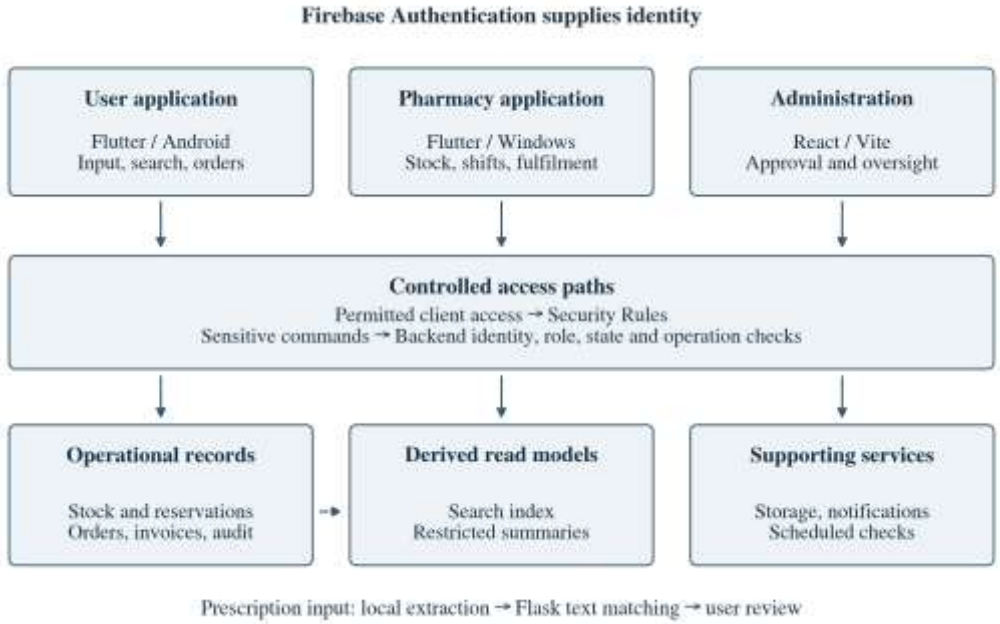


Figure 1: Application responsibilities and shared access boundaries. Operational records determine stock commitment; derived search information supports discovery.

4.3 Inventory Reservation and Repeat-safe Commands

Let S_i represent the eligible stock of item i and R_i its reserved quantity. A consistent base unit is used for package and piece sales. The available quantity A_i is given by Equation 1.

$$A_i = S_i - R_i, \text{ with } 0 \le R_i \le S_i. \quad (1)$$

Confirmation requires $q_i \le A_i$ for every requested item at the transactional read. The accepted set is reserved atomically or not at all. Confirmation increases R_i without changing S_i. Preparation reduces both by q_i, converting the reservation to a sale without deducting availability twice. Cancellation before invoicing releases the reservation.

If a previously read document changes, Firestore transactions retry and commit successful writes together (Google, n.d.-c). Application rules specify which records are coupled and which transitions are permitted.

An operation identifier links each command to its actor, action, and content fingerprint. Repeating identical content returns the recorded result; changed content under the same identifier is rejected. Lost replies therefore need not create duplicate invoices or financial effects. Downstream notification handling also accommodates repeated delivery.

4.4 Order States and Financial Transitions

Orders enter the received state without reservation. Unsupported insurance requires a decision on proceeding without coverage before review continues. The pharmacy can confirm, reject, or propose partial fulfilment. Partial offers reserve no stock. Acceptance checks the current offer version and all quantities, and a revised offer requires renewed consent.

An authorised employee with an open shift prepares a confirmed order, creating one invoice and consuming its reservation. Collection follows later (Figure 2). The first ready time recorded by the server determines the collection deadline:

$$\text{Collection deadline} = \text{first ready time} + 24 \text{ hours}. \quad (2)$$

Repeating the ready command does not extend the deadline. Expiry records an overdue condition and sends alerts while preserving the ready state. Only a successful authorised reversal changes an invoiced order to not collected. Delivery and reversal compete to finalise the order, with only one terminal outcome allowed. Pre-invoice cancellation releases reservations.

Pharmacy processing time runs from availability for review to first readiness, excluding the periods spent awaiting the user's partial-offer decision. This separates pharmacy handling time from customer decision time and from the later collection deadline.

Customer collection is assigned to the shift in which it occurs, while the invoice retains its creation shift. The insurer's share becomes an internal receivable, with payment recorded separately. Returns reduce unpaid claims or create credit after insurer payment. Delivered medicines are quarantined on return; eligible undelivered items return to their original batches.

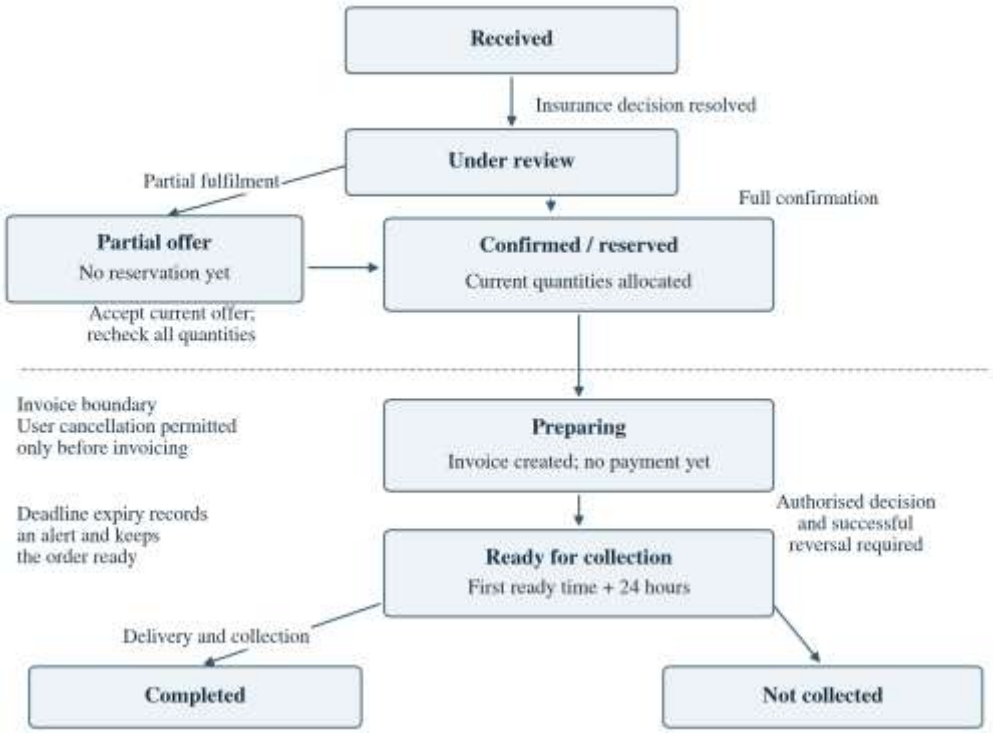


Figure 2: Principal order states and the invoice boundary. Partial-offer acceptance rechecks quantities. An overdue alert preserves the ready state; cancellation and rejection branches are omitted for clarity.

4.5 Access Boundaries and Consistency Rules

User, owner, employee, and administrator authority are checked separately. Users access their own prescriptions, beneficiaries, and orders; pharmacies receive the information needed for their orders. Administrative read models exclude private health and prescription fields. Approval, visibility, and acceptance of new orders are separate controls, preserving existing obligations when discovery is restricted.

Five consistency rules guide implementation and evaluation:

I1. Stock allocation: each accepted quantity fits current available stock; competing orders cannot allocate the same remaining quantity.

I2. Single business effect: repeated or competing commands cannot duplicate a reservation, invoice, collection, or reversal for the same operation.

I3. Authority and state: protected transitions require the proper actor, relationship, permissions, and source state; offer acceptance also requires its current version.

I4. Coupled records: stock, invoice, financial, and required audit effects commit together; interruption cannot leave only part of a transition.

I5. Terminal outcome: collection and not-collected reversal cannot both finalise an order; deadline alerts perform neither transition.

5 RESULTS

5.1 Evaluation Environment and Coverage

User tests ran on an Infinix X656 with Android 10, an eight-core 2.0-GHz processor, and 4 GB RAM. Pharmacy and administrative tests used an HP ProBook 450 G5 with an Intel Core i5-8250U, 8 GB RAM, and Windows 11 Pro. Desktop checks covered 1920 × 1080 and 1366 × 768. Administration was checked in Chrome and Edge, including emulator-based access tests; the timing checkpoint used Chrome at 1920 × 1080.

Table 3 brings together the results: 26/26 user functional and component integration cases passed, covering 34 user functional requirements, alongside 39/40 pharmacy cases and 15/15 administrative cases. All 12 executed system integration scenarios passed out of 14 planned.

Table 3: Final test outcomes by application and evaluation group.

Test group	Plan	Run	Pass	Other outcomes
User functional / integration	26	26	26	None
User critical / non-functional	10	7	7	3 without an executed result
Pharmacy functional / integration	40	40	39	1 notification failure
Pharmacy critical / non-functional	18	17	17	1 unexecuted; 3 successful retests included
Pharmacy performance	6	6	5	1 partial load-test result
Admin. functional / integration	15	15	15	None
Admin. critical / non-functional	11	11	9	2 partly exercised
Admin. performance	6	5	3	1 failed, 1 partial, 1 unexecuted
System integration	14	12	12	2 unexecuted

Plan denotes specified cases, Run denotes cases with recorded execution, and Pass denotes final recorded passes; Admin. means administration. Partial executions are included in Run and excluded from Pass. These counts describe test cases, not an overall system-quality score. The three successful retests represent final outcomes of previously defined test cases, not additional test cases.

The unexecuted integration scenarios concerned the unsupported-insurance decision and notification failure with session switching. A pharmacy visibility-change notice failed, although general administrative notices arrived. Two administrative critical cases remained partial: repeated submission lacked actual disconnection, and notification failure used a simulated provider.

5.2 Transaction Integrity and Integration Behaviour

Competing requests for the last available quantity produced one successful reservation and one rejection, without negative stock. Repeated invoice submission produced one invoice and one stock deduction. Stale search information could not authorise stock that was already reserved. These results support the allocation and repetition controls.

Partial-offer tests required a revised offer and renewed consent when availability changed between proposal and acceptance. Table 4 links these integrity challenges to the recorded outcomes.

Table 4: Selected integrity tests and observable business outcomes.

Case	Challenge	Recorded outcome
TC-PH-NF05	Two concurrent requests for the last available quantity.	One reservation succeeds; the other is rejected; stock does not become negative.
TC-PH-NF06	Repeated sale or order-to-invoice command.	One invoice and one corresponding stock deduction; no repeated financial effect.
TC-PH-NF08, retest	Internal purchase failure before commit, followed by retry.	Successful retest after adding a beforeCommit hook; linked purchase records were checked for atomicity.
TC-PH-NF12	Delivery competes with a not-collected reversal; timed processing repeats.	One terminal outcome and one set of effects; overdue processing alone preserves the ready state.
TC-ADM-NF04	Concurrent approval and rejection of one registration request.	One decision is committed; the competing decision is rejected without duplicate effects.
INT-08	Availability changes after a partial offer and before acceptance.	No allocation of changed content without a revised offer and a new user decision.

Case identifiers refer to the execution records. The results apply to the tested scenarios and failure points.

Three pharmacy corrections passed retesting. A beforeCommit hook enabled the missing internal atomicity failure test; layout and scrolling changes removed a seven-pixel finance-page overflow at 1366 × 768; and audit entries added to the atomic operation corrected missing transition records. Final critical-test counts include these retests.

Timing and accounting tests checked further effects across applications. Subtracting five hours spent awaiting user decisions from 31 elapsed hours left 26 hours of pharmacy processing. An invoice of 100 allocated 30 to the customer and 70 to insurance. Collection in a later shift preserved the invoice's creation shift while assigning payment to the collection shift. Return tests distinguished unpaid-claim reduction from credit after insurer payment.

5.3 Access Control and Privacy

Access tests rejected unauthenticated operations, cross-pharmacy violations, client-supplied role escalation, and unauthorised administrative actions. Responses excluded the private health fields planted in test data. When approval and rejection competed for one registration request, only one decision was committed, with both state and identity checked.

5.4 Performance Outcomes

Five of six pharmacy performance cases passed (Table 5). All four commands met the P95 limit. Confirmation recorded 331.29 ms at P95 and 10.94 seconds at maximum, but its 1.10% error rate exceeded the 1% ceiling by 0.10 percentage points. The other commands recorded error rates of 0.13–0.18%. The overall pharmacy HTTP load-test acceptance criteria were only partially met.

Table 5: Pharmacy HTTP results for separate five-minute command workloads.

Command	HTTP requests	Mean (ms)	P95 (ms)	Maximum (ms)	Errors (%)
confirmFullOrder	4,775	185.80	331.29	10,940	1.10
startPreparing	5,278	91.00	78.00	10,020	0.18
completeOrder	5,238	100.68	215.81	10,030	0.13
refundOrder	5,466	50.28	46.40	10,000	0.18

Latency statistics summarise all measured HTTP requests in each run. Each run reached 25 virtual users. Acceptance required P95 ≤ 3,000 ms and an error rate ≤ 1%. Maximum values are converted from the seconds displayed in the k6 output.

For confirmation, approximately 95% of measured requests completed in 331.29 ms or less. The maximum identifies the slowest request; passing a percentile threshold does not require every request to meet it. Aggregate data cannot establish the cause of the slower tail or whether the slowest request was an error.

Dashboard loading averaged 3,287.46 ms, exceeding its target by 287.46 ms (Table 6). Registration search averaged 2,872.03 ms but reached 18,474.03 ms at both P95 and maximum, showing why the mean alone is insufficient. Order search, pharmacy search, map loading, and report generation met their respective recorded targets.

Table 6: Administrative interface timings under the tested data sizes.

Operation	Mean (ms)	P95 (ms)	Maximum (ms)	Target (ms)
Dashboard loading	3,287.46	4,218.36	4,218.36	3,000
Registration search	2,872.03	18,474.03	18,474.03	3,000
Order search	1,009.02	1,844.51	1,844.51	3,000
Pharmacy search	758.77	1,034.50	1,034.50	3,000
Coverage map	2,199.02	3,200.92	3,200.92	10,000
Aggregated report	3,329.33	4,612.20	4,612.20	10,000

Dashboard and search tests used 10 repetitions; map and report tests used 5. In these small samples, the reported P95 equals the maximum under the percentile calculation used, although the two statistics remain conceptually distinct. Search routes used at least 1,000 records, the map 200 pharmacies, and the report at least 1,000 orders. Recorded operation error rates were 0%; administrative HTTP load testing was unexecuted.

5.5 Prescription Processing and Usability Observations

User-application tests recorded average prescription processing of approximately five seconds, with a slower first request after inactivity. Local reminders fired within approximately two seconds of schedule. These observations describe timing under the tested conditions; extraction accuracy and structured usability require separate assessment (Section 6.4).

The concurrent-search case had no recorded actual result. Representative user and pharmacy usability cases were not carried out; administrative task-completion observations lacked a participant count and standardised score.

6 DISCUSSION

6.1 Answers to the Research Questions

RQ1 is answered through a coordinated sequence. Reviewed input and discovery lead to current-stock confirmation; preparation creates the invoice and sale; collection records payment; and reversal resolves undelivered orders. Operation identifiers and state checks protect transitions when actions repeat or overlap. Stock, invoice, and terminal-action tests demonstrate these controls where inconsistent decisions would affect more than one application.

For RQ2, the functional results and 12 successful integration scenarios support the feasibility of the implemented workflow. The strongest evidence extends beyond interface behaviour: one allocation under competition, one invoice under repetition, renewed consent for a changed offer, and one terminal outcome. Separating P95, maximum latency, and error rate makes performance findings actionable without confusing ordinary response times with the slowest requests.

6.2 Relationship to Previous Work

Roshita connects the discovery criteria of Wondmagegn et al. (2023) and the operational stock visibility emphasised by Barrington et al. (2010) to explicit fulfilment transactions. Its engineering evidence complements these studies within a different evaluation scope. Soeno et al. (2024), Lundhaug et al. (2025), and Sattar et al. (2025) offer directions for assessing corrected-entry time, recognition errors, pharmacy effort, usability, and information quality in subsequent field work.

6.3 Implications for Pharmacy Software Design

Separating discovery indexes from authoritative stock supports efficient searches while protecting current quantities at confirmation. The interface should make this commitment point clear. An overdue alert similarly signals an outstanding order while leaving inventory and financial reversal to an authorised transaction.

Separating invoice creation, customer collection, insurer receivables, and insurer payment preserves accountability across shifts and returns. These distinctions are particularly useful when preparation and collection occur at different times. Their operational value depends on accurate starting quantities and recording counter sales as well as application orders.

6.4 Study Boundaries and Further Evaluation

The findings establish selected software properties, not clinical benefit, reduced travel, or recognition accuracy. The present evaluation did not measure OCR accuracy against verified reference transcriptions or in field use; pharmacist-validated testing of handwritten prescriptions and Arabic-language details remains future work. The survey was exploratory and used convenience recruitment with a predominantly young sample. Qualitative work covered five sites without a separate participant count or a claim of thematic saturation. Processing observations lacked sample sizes and medicine-level errors, so the mean cannot verify that 90% finish within ten seconds. Validated recognition of handwritten and Arabic prescriptions, clinical decision support, external insurer verification, and electronic payment lie outside the implementation.

Coverage is constrained by the documented cases, limited fault injection, and the tested devices and browsers. The beforeCommit retest covers one failure boundary, not all interleavings or permission combinations; independent security assessment and representative usability testing remain necessary. Local-emulator timings do not establish deployed cloud performance. Administrative samples contain only five or ten repetitions, and administrative HTTP load testing was unexecuted. With so few measurements, the reported administrative P95 values coincide with the sample maxima and are not reliable estimates of the 95th-percentile latency; broader benchmarking is planned for cloud deployment. Short local lookup and save timings lack boundaries suitable for cloud round-trip claims. Replication also requires a versioned build, request-level logs, and exact workload scripts to resolve differing descriptions of hold periods.

Engineering follow-up should address visibility-change notifications, confirmation errors and extreme delays, and dashboard and registration-search response. Remaining scenarios should test actual disconnection and live notification-provider failure. Regression tests should preserve reservation, repetition, invoicing, reversal, and privacy controls.

A supervised pilot should measure successful medicine acquisition, time to confirmed availability, cancellation and rejection reasons, stock discrepancies and reconciliation effort, staff workload, and usability across roles. Prescription evaluation should use pharmacist-verified reference names and report omissions, incorrect matches, and the time needed to correct entries.

Clinical, privacy, and institutional arrangements should be agreed with participating pharmacies before real patient data are introduced. This next stage would examine whether the demonstrated coordination remains useful during routine inventory updates and order handling.

7 CONCLUSION AND RECOMMENDATIONS

Roshita uses mobile, desktop, and web applications to connect prescription-assisted discovery with pharmacy fulfilment. Requirements from 150 questionnaire responses and five pharmacy sites informed a shared model for reservations, partial offers, invoicing, collection, reversal, and restricted oversight. Functional and integration results support its feasibility, while targeted tests demonstrate consistent allocation and single business effects under the tested concurrency and repetition scenarios.

The contribution is an evaluated connection between locating nearby pharmacies with the required medicines and managing the pharmacy's stock, sales and order operations needed to supply them. This connects the user mobile application, the pharmacy desktop application, and the administrative web application within one workflow. Distinct stock, state, and accounting controls make this connection traceable across applications. The evidence provides a foundation for focused engineering improvements and the supervised field evaluation outlined above.

LIST OF SYMBOLS AND ABBREVIATIONS

S_i: eligible stock; R_i: reserved quantity; A_i: available quantity; q_i: requested quantity of item i; P95: 95th percentile of measured latency; OCR: optical character recognition; HTTP: Hypertext Transfer Protocol; PDF: Portable Document Format; CSV: comma-separated values; RAM: random-access memory.

REFERENCES

Barrington, J., Wereko-Brobby, O., Ward, P., Mwafongo, W., & Kungulwe, S. (2010). SMS for Life: A pilot project to improve anti-malarial drug supply management in rural Tanzania using standard technology. *Malaria Journal, 9*, Article 298. https://doi.org/10.1186/1475-2875-9-298

Google. (n.d.-a). *Cloud Firestore triggers*. Cloud Functions for Firebase Documentation. Retrieved October 8, 2026, from https://firebase.google.com/docs/functions/firestore-events

Google. (n.d.-b). *Text recognition v2*. ML Kit Documentation. Retrieved October 8, 2026, from https://developers.google.com/ml-kit/vision/text-recognition/v2

Google. (n.d.-c). *Transactions and batched writes*. Firebase Documentation. Retrieved October 8, 2026, from https://firebase.google.com/docs/firestore/manage-data/transactions

Google. (n.d.-d). *Writing conditions for Cloud Firestore Security Rules*. Firebase Documentation. Retrieved October 8, 2026, from https://firebase.google.com/docs/firestore/security/rules-conditions

Grafana Labs. (n.d.). *Thresholds*. Grafana k6 Documentation. Retrieved October 8, 2026, from https://grafana.com/docs/k6/latest/using-k6/thresholds/

International Organization for Standardization & International Electrotechnical Commission. (2023). *Systems and software engineering—Systems and software Quality Requirements and Evaluation (SQuaRE)—Product quality model* (ISO/IEC 25010:2023). https://www.iso.org/standard/78176.html

International Organization for Standardization, International Electrotechnical Commission, & Institute of Electrical and Electronics Engineers. (2022). *Software and systems engineering—Software testing—Part 1: General concepts* (ISO/IEC/IEEE 29119-1:2022). https://www.iso.org/standard/81291.html

Lundhaug, K., Faxvaag, A., Stokke, R., & Andreassen, H. K. (2025). Collaboration routines and workflows in a national electronic prescription system: Qualitative study. *Journal of Medical Internet Research, 27*, Article e56558. https://doi.org/10.2196/56558

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems, 24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Sattar, A., Rehman, H., Naveed, S., Khadim, S., Khan, N., Kazi, A. F., Syed, W., Al-Rawi, M. B. A., & Jamshed, S. (2025). Trustworthiness of web-based pharmacy apps in Pakistan based on the Mobile App Rating Scale: Content analysis and quality evaluation. *JMIR mHealth and uHealth, 13*, Article e59884. https://doi.org/10.2196/59884

Soeno, S., Liu, K., Watanabe, S., Sonoo, T., & Goto, T. (2024). Development of novel optical character recognition system to reduce recording time for vital signs and prescriptions: A simulation-based study. *PLOS ONE, 19*(1), Article e0296319. https://doi.org/10.1371/journal.pone.0296319

World Health Organization. (2019). *WHO guideline: Recommendations on digital interventions for health system strengthening*. https://www.who.int/publications/i/item/9789241550505

Wondmagegn, M. T., Nabaasa, E., Melesse, E. A., & Kassaw, E. A. (2023). A framework for locating prescribed medication at pharmacies. *Integrated Pharmacy Research and Practice, 12*, 127–136. https://doi.org/10.2147/IPRP.S415674