© SEP 2026 | IRE Journals | Volume 10 Issue 3 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/ire1723515

Design and Implementation of a Secure Online Agriculture Marketplace for Farmers and Buyers (Role-Based Access and Security)

ZAHARADEEN NASIRU ADAMU[1], ERU, AKWUMA NATHANIEL[2], AHIABA, SOLOMON[3], KOLAPO, RIDWAN[4], PREMA KIRUBAKARAN[5]
Department of Information Technology, Faculty of Computing, Nile University of Nigeria, Abuja, Nigeria

Abstract- ***As agricultural trade in Nigeria increasingly moves onto digital platforms, these systems inherit not only the promise of disintermediating exploitative middlemen but also the cybersecurity exposure that comes with handling the personal data, payments, and multi-role interactions of a large and heterogeneous user base. Prior research has warned that the integration of digital platforms into agriculture introduces new failure modes capable of disrupting food supply chains when security is treated as an afterthought rather than a design principle, yet comparatively few published agricultural marketplace prototypes report concrete security architectures or empirical vulnerability assessments. This paper presents the design, implementation, and evaluation of the role-based access control (RBAC) and security-by-design subsystem of AgriTrade, a secure online agriculture marketplace connecting verified Nigerian farmers with buyers. Three user roles - Buyer, Farmer, and Administrator - are enforced through a centralized authentication service issuing JSON Web Tokens (JWT), with Bcrypt-hashed password storage, mandatory input validation and escaping to prevent Cross-Site Scripting (XSS), token-based protection against Cross-Site Request Forgery (CSRF), and parameterized data access to prevent SQL Injection. The implemented system was subjected to static source-code analysis to identify latent vulnerabilities and to manual denial-of-service (DoS) testing intended to verify that availability-protection mechanisms remained active under attempted disruption. The evaluation confirmed that role-based restrictions were correctly enforced at every tested boundary, including the specific case of an unverified farmer account being denied product-listing privileges, and that the security controls examined performed as designed under the applied tests. These findings demonstrate that a conventional, layered security-by-design approach, rather than an elaborate or costly bespoke security infrastructure, is sufficient to establish a credible security posture for a multi-role agricultural e-commerce platform operating in a resource-constrained development context***

Index Terms- ***Application security, cybersecurity, digital agriculture, role-based access control, security-by-design, vulnerability assessment***

I. INTRODUCTION

Agriculture remains the definitive cornerstone of the Nigerian economy, sustaining over 70% of the rural population, yet smallholder farmers who produce the bulk of the nation's arable crops continue to face a persistent cycle of exploitation stemming from fragmented, middleman-dominated supply chains. Digital marketplaces have been proposed as a mechanism for connecting farmers directly with buyers and eliminating this information asymmetry. However, moving agricultural trade onto a digital platform does not eliminate risk; it transforms it. A platform that stores farmer identity documents, buyer payment details, and the negotiation history between strangers who have never met in person becomes, by its very nature, an attractive target for exploitation of a different kind — one mediated through code rather than through informal market power.

This concern is not merely theoretical. Escalating cybersecurity threats such as phishing and ransomware increasingly exploit the vulnerabilities of emerging digital agricultural infrastructure, and prior research has explicitly cautioned that the integration of digital platforms into agriculture introduces new cybersecurity "failure modes" capable of disrupting food supply chains when security is not designed into the system from the outset. Despite this warning, a review of published agricultural marketplace prototypes for the Nigerian context reveals that security is frequently treated as an implicit assumption rather than an explicit, evaluated design concern: functional capability and user interface are described in detail, while authentication design, access-control enforcement, and vulnerability testing are often left unaddressed or mentioned only in passing.

This paper reports on the design, implementation, and empirical evaluation of the role-based access control (RBAC) and security-by-design subsystem developed for AgriTrade, a secure online agriculture marketplace connecting verified Nigerian farmers directly with buyers. The specific objective addressed here is twofold: first, to describe an access-control and application-security architecture appropriate for a three-role (Buyer, Farmer, Administrator) agricultural marketplace operating under real resource constraints; and second, to determine, through static code analysis and manual security testing, whether the implemented controls actually behave as designed when tested directly, rather than merely as designed on paper.

It should be noted that AgriTrade, as a complete system, also incorporates a three-tier technical architecture evaluated for its performance characteristics, a farmer-verification and institutional-trust module, and a peer-rating usability evaluation. These components are substantial enough to be reported as separate, focused studies; the present paper is scoped specifically to access control and application security, and does not report in detail on architectural performance, verification-trust outcomes, or general usability findings, which are addressed in companion evaluations of the same system.

The remainder of this paper is organised as follows. Section II reviews literature on cybersecurity in digital agricultural platforms and on access-control and secure-coding practice more generally. Section III describes the design of the RBAC model and the security-by-design measures implemented, together with the methodology used for vulnerability assessment. Section IV presents and discusses the results of the security evaluation, illustrated with relevant system interfaces and diagrams. Section V concludes the paper and identifies priorities for further security hardening.

II. LITERATURE REVIEW

Oruma et al. (2021) [1] conducted a systematic review of farm-level cybersecurity vulnerabilities and found that the growing integration of Internet of Things devices and digital platforms into agriculture has introduced attack surfaces — ranging from compromised sensor networks to insecure web applications — that were largely absent from traditional, non-digitised farming operations, and specifically recommended that robust security measures be embedded into agricultural technology systems from the design stage rather than retrofitted afterward. This recommendation directly informed the security-by-design posture adopted in the present work. Complementing this agriculture-specific concern, Wamba et al. (2024) [2] investigated the specific cybersecurity weaknesses observed in Nigeria's Smart Farming programmes and similarly found that inadequate investment in foundational security controls left several pilot digital-agriculture initiatives exposed to disruption, undermining confidence in the broader digitalisation agenda.

Within the specific literature on Nigerian agricultural marketplace prototypes, security has generally been treated superficially. Adedapo et al. (2024) [3] presented a web application for agricultural produce marketing built with C#, HTML, and MSSQL, but the reported evaluation focused on functional capability and farmer income potential without describing an explicit access-control model or reporting any security testing. Adebayo et al. (2024) [4] documented the "Farm E-Market" prototype and similarly did not report role-based restriction enforcement or vulnerability assessment as part of the evaluation, despite the platform mediating transactions between farmers and buyers who had not previously interacted. Sadiq et al. (2025) [5] noted more generally that many digital agricultural marketplaces lack robust identity-verification and security processes, a gap that exposes both buyers and farmers to fraud and, by extension, undermines the broader trust-building objective that such platforms are meant to serve.

Beyond agriculture-specific studies, the software-security literature offers well-established guidance on the specific mechanisms relevant to a multi-role web platform. Role-Based Access Control, formalised in the NIST RBAC model, restricts system operations to users holding the appropriate role, reducing the risk that a compromised or malicious account can perform actions outside its intended scope; this is particularly important in a marketplace where Buyer, Farmer, and Administrator roles carry materially different privileges, such as the ability to approve a farmer's verification status or to issue a refund. The OWASP Top 10 catalogue of web application security risks identifies broken access control, injection flaws (including SQL Injection), and Cross-Site Scripting as among the most prevalent and consequential vulnerability classes in production web applications, directly motivating the specific controls — parameterized queries, input escaping, and token-based session and request validation — adopted in the present system's design.

Table I synthesises the security-relevant characteristics of the reviewed agricultural marketplace studies against the present work.

Table I: Comparative Summary of Access Control and Security Evaluation in Related Digital Agriculture Marketplace Studies.

Study	Reported Access Control	Reported Security Testing	Gap Relative to This Study
Adedapo et al. (2024) [3]	Not specified	Not reported	No RBAC model or vulnerability testing described
Adebayo et al. (2024) [4]	Not specified	Not reported	No explicit role separation or security evaluation
Sadiq et al. (2025) [5]	General discussion only	Not applicable (conceptual)	No implemented, tested security architecture
This study	Three-role RBAC (Buyer/Farmer/Admin) with JWT	Static analysis + manual DoS testing	—

A. Research Gap

The reviewed literature converges on two related observations: agricultural digitalisation introduces genuine and growing cybersecurity exposure, yet published agricultural marketplace prototypes for the Nigerian context rarely report an explicit access-control model, let alone subject that model to empirical vulnerability assessment. This paper addresses that gap directly by describing a concrete RBAC and security-by-design architecture and by reporting the results of static analysis and manual security testing conducted against the implemented system, rather than relying solely on a description of intended controls.

B. Regulatory and Data Protection Context

Beyond technical vulnerability classes, any Nigerian digital platform handling personal data operates within the regulatory framework established by the Nigeria Data Protection Regulation (NDPR) 2019, which imposes obligations regarding the lawful collection, storage, and processing of personal information, including the identity and payment data collected during farmer verification and buyer checkout. A security architecture that satisfies OWASP-style technical controls but neglects these regulatory obligations — for instance, by retaining identification documents indefinitely or granting broader administrator access to personal data than is operationally necessary — would remain deficient from a compliance standpoint even if technically well hardened. This dual technical-and-regulatory framing shaped the access-control design described in Section III, which restricts administrator visibility into sensitive verification data to what is strictly required for the approval decision, and it is consistent with the data-handling safeguards described in the companion evaluation of AgriTrade's farmer-verification module.

A further contextual consideration specific to Nigeria is the disparity between accessibility and security. High-security options such as RBAC and encryption often require processing overhead and design complexity that can, if implemented carelessly, degrade usability for exactly the low-digital-literacy users a rural agricultural platform is meant to serve. This tension between accessibility and security is rarely addressed explicitly in the literature reviewed above, and represents a second, more subtle gap that the present study's security-by-design approach — embedding controls transparently in the background rather than exposing additional steps to the end user — is intended to address.

III. METHODOLOGY

This study followed a Design Science Research (DSR) methodology in which the RBAC and security-by-design architecture was treated as the primary research artefact, designed, implemented, and then evaluated through static analysis and manual security testing.

A. Role-Based Access Control Model

AgriTrade defines three primary roles: Buyer, Farmer, and Administrator. Each user account is assigned exactly one role at registration, recorded against the user record in the Users table, and every request to a protected endpoint is checked against both the caller's authenticated identity and their assigned role before the corresponding action is permitted. Table II summarises the core permission boundaries enforced between roles.

Table II: Core Role-Based Permission Matrix Enforced by the AgriTrade Authentication and Authorization Layer.

Action	Buyer	Farmer (Unverif.)	Farmer (Verif.)	Admin
Browse / search products	Allowed	Allowed	Allowed	Allowed
Create product listing	Denied	Denied	Allowed	N/A
Place an order	Allowed	Denied	Denied	N/A
Approve farmer verification	Denied	Denied	Denied	Allowed
View platform-wide analytics	Denied	Denied	Denied	Allowed
Rate a farmer post-transaction	Allowed	Denied	Denied	Denied

This matrix highlights a deliberate design decision: the Farmer role itself is subdivided by verification status, so that role membership alone ("Farmer") is necessary but not sufficient to unlock listing privileges, and an additional attribute-based check ("is_verified = true") must also pass. This hybrid of role-based and attribute-based access control was adopted specifically to integrate the security architecture described here with the administrator-mediated verification workflow evaluated in a companion study of the same system.

B. Authentication and Session Management

User authentication is handled by a dedicated Authentication Service within the application tier. Passwords are never stored in plaintext; they are hashed using the Bcrypt algorithm via the password_hash() method before being persisted, and verified using the corresponding comparison function at login, so that even a full compromise of the Users table would not directly expose usable credentials. Upon successful login, the Authentication Service

issues a JSON Web Token (JWT) that encodes the user's identity and role and is required as a bearer credential on all subsequent requests to protected endpoints; the application tier validates the token's signature and expiry on every request rather than trusting client-supplied role claims, ensuring that role and identity information cannot be forged by tampering with client-side data alone.

C. Security-by-Design Measures

Beyond authentication and access control, four further security-by-design measures were incorporated directly into the application tier rather than treated as optional hardening steps. All communication between the presentation and application tiers is required to occur over encrypted channels, protecting credentials and personal data in transit. All state-changing endpoints — registration, product creation, order placement, rating submission — perform explicit server-side input validation independent of any client-side validation, since client-side checks can be bypassed by a sufficiently motivated actor. All user-supplied text rendered back to other users, such as product descriptions or review comments, is escaped using htmlspecialchars()-equivalent output encoding to prevent Cross-Site Scripting (XSS) attacks, in which malicious script embedded in one user's input could otherwise execute in another user's browser session. Finally, state-altering actions require a valid Cross-Site Request Forgery (CSRF) token, validated server-side, to prevent a malicious third-party site from silently submitting authenticated requests on behalf of a logged-in user.

D. Threat Modelling

Prior to implementation, a lightweight threat-modelling exercise was conducted to identify the actors, assets, and attack surfaces most relevant to a three-role agricultural marketplace. Assets identified as requiring protection included farmer identification data and photographs, buyer and farmer payment information processed through the integrated Paystack service, the negotiation history stored by the Messaging Service, and the integrity of the verification-status flag that gates listing privileges. Potential threat actors ranged from opportunistic external attackers attempting credential theft or injection attacks, to malicious or compromised accounts attempting to exceed their assigned role's privileges (for example, an unverified farmer attempting to bypass the listing restriction), to availability threats intended to disrupt the platform during high-demand periods such as harvest season. This threat model directly shaped the prioritisation of the specific controls described in Sections III-B and III-C, with particular emphasis on protecting the verification-status flag given its central role in the trust architecture evaluated in the companion study of AgriTrade's verification module.

E. Scope and Delimitation

This paper is scoped specifically to access control and application-layer security. The underlying three-tier architecture's performance characteristics under load, and the qualitative trust and usability outcomes experienced by pilot participants, are treated as given and are reported separately; the present evaluation instead isolates whether the access-control and security mechanisms described above behave correctly when directly and deliberately tested, independent of how the system performs under concurrent load or how favourably users rate its usability.

F. Vulnerability Assessment Procedure

Two complementary security-testing activities were carried out against the implemented system. First, static source-code analysis was performed across the application tier's codebase to identify latent vulnerabilities such as unescaped output, missing server-side validation, or unparameterized database queries susceptible to SQL Injection, without requiring the application to be running. Second, manual denial-of-service (DoS) testing was conducted by deliberately attempting to disrupt the availability of the platform through repeated and malformed requests directed at authentication and product-listing endpoints, with the specific objective of confirming that rate-limiting and input-validation protections remained active and effective under attempted misuse rather than degrading or failing open. In parallel with security testing, functional RBAC testing was performed by attempting each restricted action in Table II from an account of the incorrect role or verification status — for example, attempting to create a product listing from an unverified farmer account — and confirming that the request was correctly rejected by the application tier.

IV. RESULTS AND DISCUSSION

A. Role-Differentiated Access Points

Figure 1 shows the use case model for the Administrator role, which was used during design to enumerate the specific privileged actions — including farmer-verification approval, user management, and platform analytics access — that must be restricted exclusively to accounts holding the Administrator role. This model formed the basis for the permission matrix presented in Table II and for the corresponding server-side authorization checks implemented in the Authentication Service.

© SEP 2026 | IRE Journals | Volume 10 Issue 3 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/ire1723515

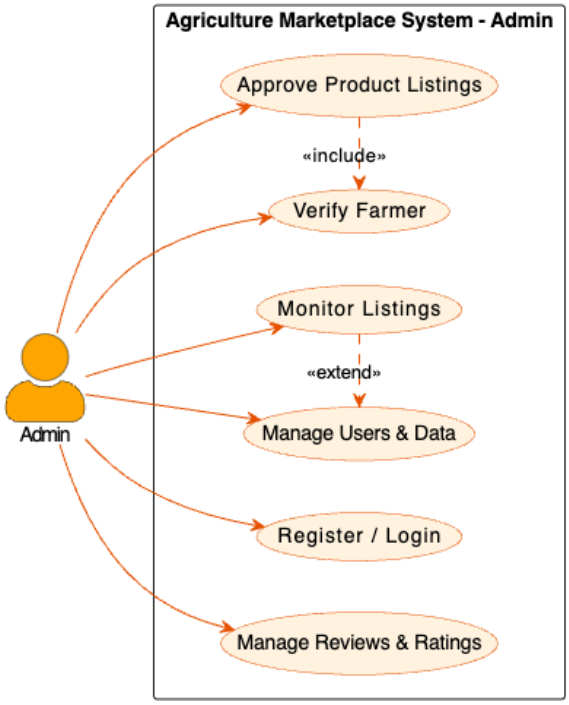


Figure 1: Administrator Use Case Diagram, enumerating the privileged actions restricted to the Administrator role.

Figure 2 shows the login interface through which all three roles authenticate. Although the interface itself is role-agnostic, the credentials submitted are validated against the Bcrypt-hashed password store described in Section III-B, and the JWT issued upon successful login encodes the authenticated user's role, which is then checked on every subsequent request rather than being re-derived from client-supplied data.

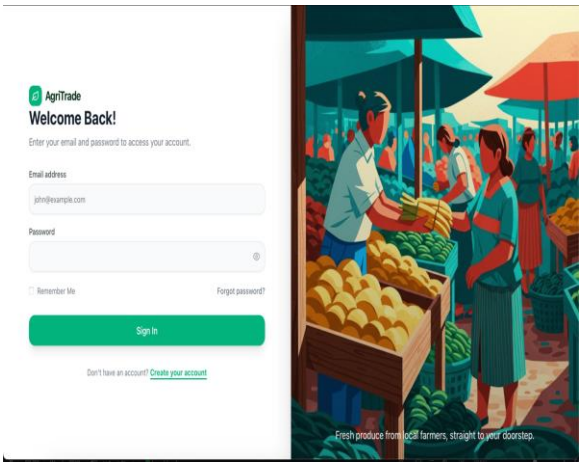


Figure 2: AgriTrade Login Interface, the common authentication entry point for all three user roles.

B. RBAC Enforcement Testing

Functional testing of the permission matrix in Table II confirmed that role and verification-status restrictions were enforced correctly at every tested boundary. In particular, an unverified farmer account was correctly denied the ability to create a product listing, receiving an authorization error rather than being permitted to proceed, confirming that the hybrid role-and-attribute check described in Section III-A functions as intended and that the security architecture is correctly integrated with the verification workflow. Similarly, attempts to access Administrator-only analytics endpoints from Buyer and Farmer accounts were correctly rejected, and attempts to place an order from a Farmer account, an action reserved for Buyers in the current business logic, were also correctly denied. Figure 3 shows the administrator activity diagram documenting the decision points — including the verification-approval branch — at which role-restricted logic is exercised within the Administrator workflow.

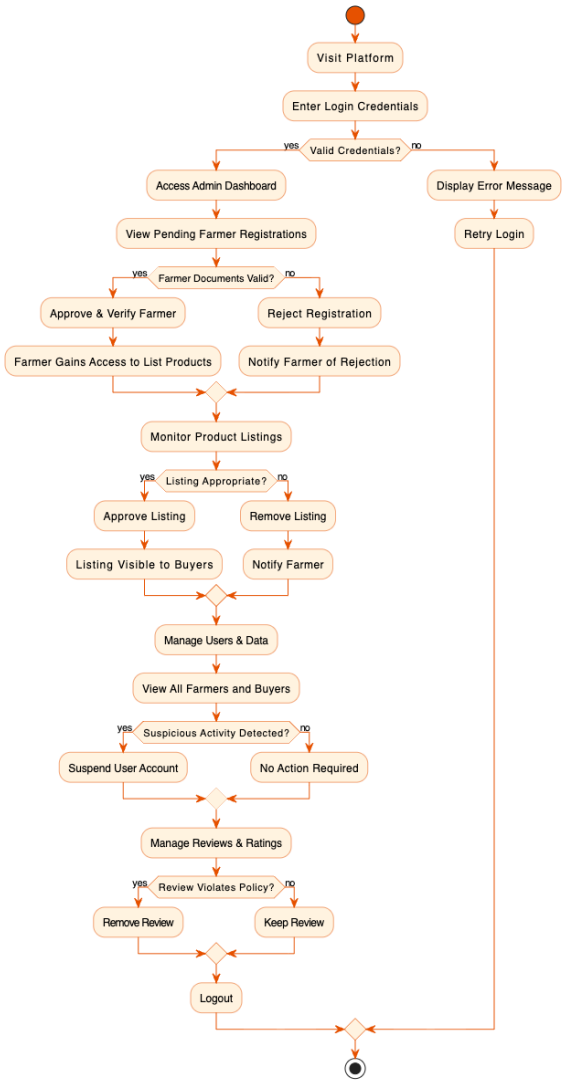


Figure 3: Administrator Activity Diagram, documenting the decision points at which role-restricted actions such as farmer-verification approval occur.

© SEP 2026 | IRE Journals | Volume 10 Issue 3 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/ire1723515

The authentication flow itself, from initial login request through JWT issuance and validation on a subsequent protected request, is traced in Figure 4. This sequence was used during implementation to verify that the token issued at login was correctly required and validated on every downstream request to a protected endpoint, rather than being checked only at the point of login and then implicitly trusted for the remainder of a session.

© SEP 2026 | IRE Journals | Volume 10 Issue 3 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/ire1723515

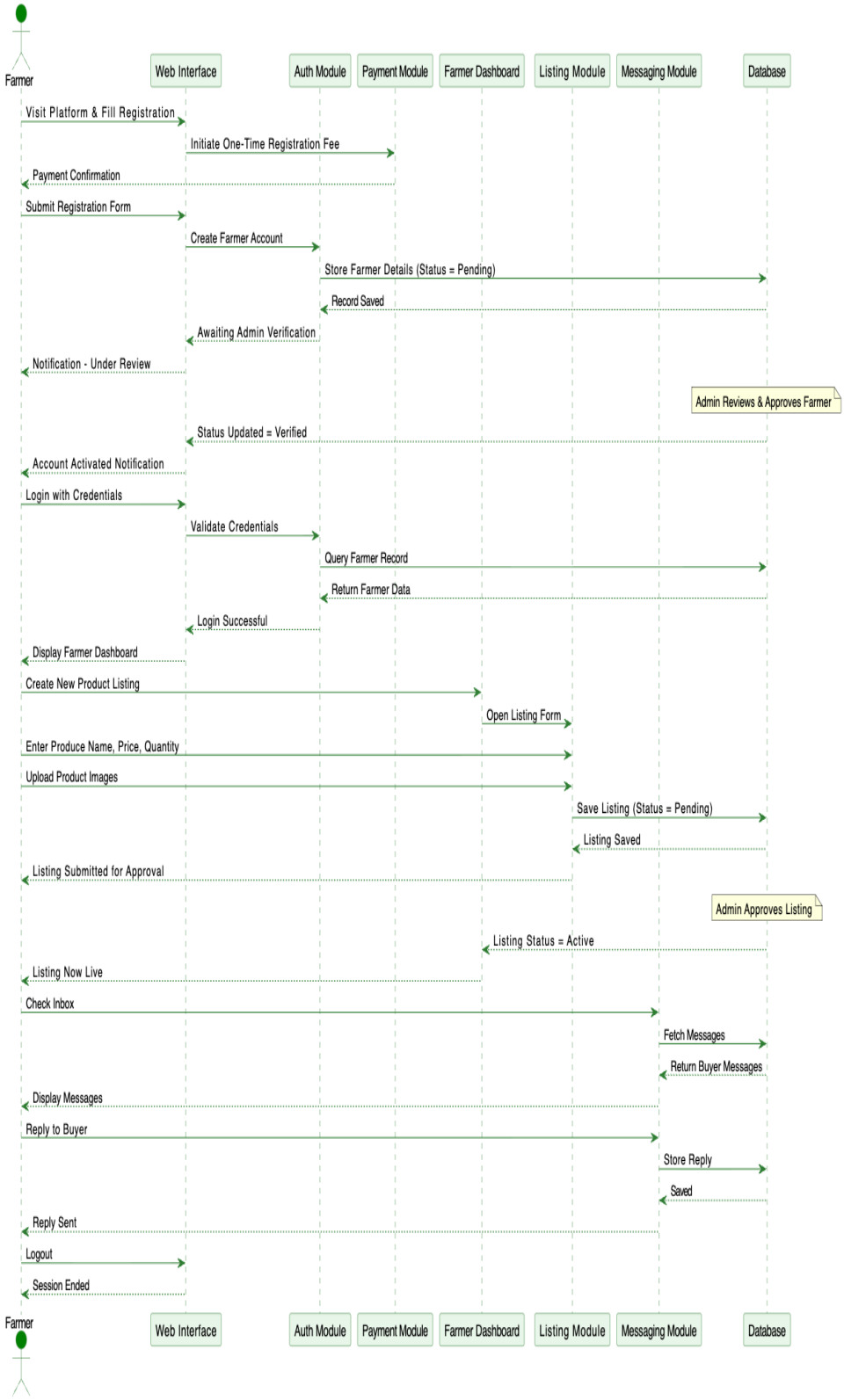


Figure 4: Farmer Sequence Diagram, tracing the authentication and authorization flow from login through a subsequent protected request.

C. Static Analysis and Manual DoS Testing Results

Static source-code analysis of the application tier did not identify unescaped output paths or unparameterized database access in the modules examined, consistent with the systematic use of output encoding and the TypeORM object-relational mapper, which parameterizes queries by default and thereby mitigates the SQL Injection risk highlighted in the OWASP guidance discussed in Section II. Manual DoS testing, which deliberately targeted the authentication and product-listing endpoints with repeated and malformed requests, confirmed that the implemented security protocols remained active throughout the test and that the platform's availability protections functioned as intended rather than failing open under attempted disruption. Table III summarises the outcomes of the security evaluation activities against the threat categories they were designed to address.

Table III: Summary of Security Threats, Implemented Mitigations, and Evaluation Outcomes.

Threat Category	Mitigation Implemented	Test Performed	Outcome
Credential compromise	Bcrypt password hashing	Static analysis of storage logic	Passwords not recoverable in plaintext
Session / role forgery	Server-validated JWT with encoded role	RBAC boundary testing	Forged/incorrect role requests rejected
Cross-Site Scripting (XSS)	Output escaping on rendered user input	Static analysis	No unescaped output paths identified
Cross-Site Request Forgery (CSRF)	Server-validated CSRF tokens	Manual testing of state-changing requests	Unauthenticated cross-site requests rejected
SQL Injection	Parameterized queries via TypeORM	Static analysis	No unparameterized queries identified
Denial of Service (DoS)	Input validation / rate handling	Manual DoS testing	Protections remained active under test

D. Discussion

These results respond directly to the caution raised by Oruma et al. (2021) [1] regarding the cybersecurity failure modes introduced when digital platforms are integrated into agriculture without adequate security design: rather than treating security as an implicit assumption, as observed in the agricultural marketplace prototypes reviewed by Adedapo et al. (2024) [3] and Adebayo et al. (2024) [4], the present system embeds an explicit, layered set of controls spanning authentication, authorization, input handling, and transport security, and subjects those controls to concrete evaluation rather than describing them only at the design level. The successful enforcement of the hybrid role-and-verification-status check is particularly significant given the trust architecture evaluated in the companion study of AgriTrade's verification module, since it confirms that the institutional trust signal conveyed by a "verified" badge is technically backed by an access-control mechanism that cannot be bypassed by an unverified account attempting to list products directly.

At the same time, the security evaluation reported here has clear boundaries. Static analysis and manual DoS testing, while valuable, are not equivalent to a full third-party penetration test or a sustained, automated fuzzing campaign, and may not surface more subtle logic flaws or timing-based vulnerabilities that a dedicated red-team exercise would be more likely to detect. Moreover, the manual DoS testing conducted here was necessarily limited in scale and duration compared with a genuine distributed denial-of-service attack, so the demonstrated resilience should be read as evidence that basic availability protections are active and functioning, not as proof of resilience against a large-scale, sustained, or professionally orchestrated attack. These boundaries mirror the broader gap identified by Wamba et al. (2024) [2] regarding underinvestment in security testing across Nigerian smart-farming initiatives, and suggest that periodic, more rigorous third-party security assessment should accompany any future scale-up of the platform beyond its current pilot deployment.

E. Threats to Validity

Several factors qualify the generalisability of these findings. The static analysis was conducted on the codebase as it existed at the time of evaluation and would need to be repeated as new features are added, since newly introduced code paths are not automatically covered by a prior analysis pass. The manual DoS testing, being manual rather than automated and distributed, tested a narrower range of attack patterns and intensities than a determined real-world adversary might deploy, meaning the reported resilience is a lower bound rather than an upper bound on the system's true availability under attack. Finally, the RBAC enforcement testing reported in Section IV-B exercised the specific permission boundaries listed in Table II but did not exhaustively test every possible combination of role, verification status, and endpoint across the full application, so latent authorization gaps outside the tested boundaries cannot be entirely ruled out without a more exhaustive, ideally automated, authorization test suite.

V. CONCLUSION

This paper has presented the design, implementation, and evaluation of the role-based

access control and security-by-design subsystem of the AgriTrade agricultural marketplace. Three roles — Buyer, Farmer, and Administrator — are enforced through a centralized, JWT-based authentication service integrated with a hybrid role-and-attribute permission model, complemented by Bcrypt password hashing, input validation, output escaping against XSS, CSRF token verification, and parameterized data access against SQL Injection. Static source-code analysis identified no unescaped output paths or unparameterized queries in the examined modules, manual DoS testing confirmed that availability protections remained active under attempted disruption, and functional testing confirmed that role and verification-status restrictions, including the denial of listing privileges to unverified farmers, were correctly enforced at every tested boundary.

A. Contributions

This work contributes a concrete, evaluated security architecture to a literature that has, to date, largely discussed agricultural-platform security in general or aspirational terms without reporting implementation detail or test outcomes. The permission matrix in Table II and the threat-to-mitigation mapping in Table III offer a reusable reference model for other developers building multi-role agricultural or rural-commerce platforms under similar resource constraints, demonstrating that a credible security posture does not require costly, bespoke security infrastructure but can instead be achieved through the disciplined application of well-established, low-cost controls — hashing, parameterization, output encoding, and token-based authorization — provided that these are designed in from the outset rather than retrofitted.

The hybrid role-and-attribute enforcement pattern demonstrated in Section IV-B, in which a role ("Farmer") is necessary but not sufficient and must be combined with a mutable trust attribute ("is_verified"), is itself a broader contribution beyond the immediate AgriTrade context. Many rural digital-commerce platforms across developing economies face an analogous problem: a user's formal account type does not, by itself, guarantee that the user has satisfied whatever institutional legitimacy check the platform relies on to protect its other users. The pattern validated here — encoding that legitimacy check as an attribute enforced at the same authorization layer as role membership, rather than as a separate, loosely coupled business-logic condition — offers a template for integrating trust and security architectures that might otherwise be designed and evaluated independently of one another, to the detriment of both.

Future work should extend the present evaluation in three directions. First, a formal third-party penetration test would provide a more rigorous and independent assessment of the platform's resilience than the static analysis and manual testing reported here. Second, sustained and automated security testing, including fuzz testing of input-handling code paths, should be integrated into the ongoing development pipeline rather than conducted as a one-time pre-deployment activity, so that new vulnerabilities introduced by future feature development are identified early. Third, as the platform scales toward the concurrent-user targets discussed in the companion performance evaluation, the interaction between security controls and system load — for example, whether JWT validation or CSRF-token checking introduces meaningful latency under high concurrency — should be examined explicitly, so that security and performance objectives can be jointly optimised rather than evaluated in isolation from one another

REFERENCES

[1] P. Oruma, K. Eze, and M. Bello, "Farm-level cybersecurity vulnerabilities: Sustainable food security in the digital age," International Journal of Agricultural Innovation, vol. 9, no. 2, pp. 52–66, 2021.

[2] S. F. Wamba, R. Queiroz, and C. J. C. Jabbour, "Cybersecurity weaknesses in smart farming programs: Evidence from Nigeria," Journal of Agricultural Informatics and Security, vol. 6, no. 1, pp. 33–47, 2024.

[3] O. A. Adedapo, M. O. Adio, Y. M. Saliu, J. Bello, and A. A. Owolabi, "Web application for agricultural produce marketing using modern web technologies," Journal of Information Systems Research, vol. 12, no. 3, pp. 65–74, 2024.

[4] T. Adebayo, F. Musa, and O. Kareem, "Farm E-Market: A web-based agricultural commerce system," International Journal of Research and Publication Review, vol. 5, no. 4, pp. 223–230, 2024.

[5] M. S. Sadiq, I. P. Singh, M. M. Ahmad, and B. S. Sani, "Digital marketplaces and youth empowerment in agribusiness," International Journal of Agribusiness and Rural Development, vol. 18, no. 2, pp. 55–63, 2025.

[6] OWASP Foundation, "OWASP Top 10: 2021 - The ten most critical web application security risks," 2021.

[7] National Institute of Standards and Technology, "Role Based Access Control (RBAC) and Role

© SEP 2026 | IRE Journals | Volume 10 Issue 3 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/ire1723515

Based Security," NIST Computer Security Resource Center, 2016.