Community Engagement & Emergence Journal
Volume 7 Nomor 3, Tahun 2026
Halaman : 3043-3055

Template-Grounded Generative AI for Official Correspondence: Design and Functional Evaluation of the Smart Correspondence Assistant

Rayi Dwipanilih[1], Wida Aristanti[2], Muhammad Ikhwan[3], Suherdi[4], Rizki Firdausi Rachma Dania[5]
Digital Office Administration Study Program, Faculty of Economics and Business, Universitas Negeri Jakarta, Indonesia
[1]rayidwp@unj.ac.id*; [2]widaaristanti@unj.ac.id; [3]m_ikhwan@unj.ac.id; [4]suherdi@unj.ac.id; [5]rachmadania@unj.ac.id

Abstract
Official correspondence in professional associations often relies on copied files and separate approval records, producing inconsistent structure and weak traceability. General-purpose generative artificial intelligence can accelerate drafting but may fabricate factual content and vary mandatory document elements. This study designs and functionally evaluates the Smart Correspondence Assistant (SCA), a web-based artifact that uses PermenPANRB 21/2021 as a structural reference while separating deterministic document assembly from model-generated narrative. Following design science research, seven design requirements were derived from practitioner experience, organisational letter examples, and the reference standard. The artifact supports seven letter types, configurable multi-level approval, deterministic numbering, role-based access control, audit logging, and PDF generation. Functional evaluation comprised fifty-four scenarios across six iterative runs in an isolated PHP 7.4.33 environment. Eight defects were identified and all were remediated and retested. In the final cumulative state, all fifty-four scenarios passed: forty-six were decided through objective functional checks and eight semantic or visual scenarios were verified manually by the first author using preserved evidence packages. Because the manual reviewer was part of the development team, the result represents developer-executed functional conformance rather than independent validation. The study contributes a transferable pattern for constraining generative artificial intelligence in official correspondence by keeping structural and authority controls at the application layer, while treating generated narrative as reviewable content rather than a final document.
Keywords: Design Science Research, Generative Artificial Intelligence, Official Correspondence, Template-Grounded Generation, Functional Evaluation.

1. Introduction

Professional associations and administrative units in higher education produce a steady volume of official letters throughout the year, including requests, invitations, notifications, cooperation offers, certifications, assignments, and circulars. Drafting is generally performed by hand, by copying a previous letter file and editing the parts that change. Numbering is recorded in a logbook or a spreadsheet kept separately from the letter files themselves. Approval is requested through online chat or by circulating printed drafts, so the record of who approved what and at what time is not stored in any single place. This practice produces formatting differences between drafters, repeated revision cycles, and difficulty in retrieving earlier letters. It persists even as Indonesian public administration undergoes broad digital reform, since attention has concentrated on service delivery platforms rather than on the internal document work that supports them [1], [2].

The public availability of large language models has changed this situation within a short period. Administrative staff can now draft letters through general-purpose chat services without any supporting system. That convenience introduces a problem specific to official correspondence. General-purpose chat services do not guarantee

the completeness of structural elements required by regulation, do not issue document numbers, do not retain approval records, and may produce statements that did not originate from the drafter's input, a failure mode documented across generation tasks [3]. In official correspondence, output that reads fluently but contains incorrect data or a deviant structure carries greater risk than manual drafting, because such errors are difficult to detect on a cursory reading. Evaluations in public administration report the same pattern, finding that model-drafted documents read well while remaining weaker on factual and contextual accuracy than human-written equivalents [4].

This study concentrates on the design, construction, and functional evaluation of a software artifact rather than on measuring effectiveness across a user population. The scope is limited to outgoing Indonesian-language official letters covering the seven letter types used by Asosiasi Sarjana dan Profesi Pendidikan Administrasi Perkantoran Indonesia (ASPAPI). Selected structural provisions of PermenPANRB 21/2021 are used as a reference standard [17]; the study does not claim that the regulation directly governs ASPAPI. Signing remains a wet-ink activity performed outside the system, so electronic signatures fall outside the scope. The study also does not compare performance across language models; the model is treated as a replaceable component within the design.

On this basis, the study aims to formulate design requirements for a generative-AI-assisted official correspondence drafting tool, to design and build an artifact that keeps structural elements deterministic while only the narrative portion is model-generated, to demonstrate that artifact on a complete letter drafting scenario, and to evaluate the fulfilment of the design requirements through scenario-based functional testing.

2. Literature Review and Problem Statement

Advances in Transformer-based language models have rapidly improved the fluency and coherence of generated text, extending their use to summarisation, document drafting, and data-to-text generation. At the same time, the comprehensive survey by Ji et al. [3] shows that deep-learning-based generation is prone to hallucination, that is, text unsupported by the source input, and that the problem cuts across tasks and is not resolved by model scaling alone. In free-form drafting, hallucination degrades output quality. In official correspondence the consequences are heavier, because a signed letter carries administrative and legal effect, while errors in an institution name, a date, or a regulatory citation that do not match the input are hard for a reader to notice without a line-by-line check. The phenomenon is well documented at task level: Maynez et al. [5] found through large-scale human evaluation that abstractive summarisation models frequently produce content unfaithful to the source document, and that standard automatic metrics correlate poorly with faithfulness. Work on document drafting in the public sector points in the same direction, reporting that fine-tuned models improve language quality and tone while factual accuracy and domain coherence remain the binding constraint, and concluding that a hybrid arrangement in which model output is reviewed and refined by human experts is the more defensible configuration [4]. This matters because digitally induced change in the public sector is itself uneven, and reviews of that literature find that transformation claims frequently outrun demonstrated capability at the level of concrete work processes [6], [7].

The information systems field provides an established framework for artifact-oriented research. Hevner et al. [8] set out a conceptual framework and seven guidelines for design science research, emphasising that knowledge is obtained through building and applying artifacts and that evaluation is inseparable from the contribution. Peffers et al. [9] complement this with a six-activity process model that serves both as an execution framework and as a presentation framework. Gregor and Hevner [10] further distinguish levels of artifact abstraction that may constitute a design science contribution, and position an instantiation as a legitimate contribution when its design rationale is made explicit. These references place the artifact, rather than hypothesis testing, as the primary research output, which matches the nature of this study, which stops at functional evaluation. A parallel discussion concerns the role assigned to the human reviewer. Laux [11] argues that oversight provisions serve accountability and legitimacy as well as error correction, while noting that empirical evidence casts doubt on how reliably human overseers discharge that function. Green [12] makes the stronger claim that oversight requirements can legitimise deployment of faulty systems without addressing their underlying defects when overseers are not genuinely positioned to detect and reverse a faulty output. Both cautions bear directly on official correspondence, where the drafter and the approving officer are the only points at which a fabricated statement can be intercepted before signature, and where an approving officer under time pressure may not examine narrative text line by line.

A third strand concerns how far model behaviour can be constrained by instruction alone. Surveys of controllable text generation catalogue mechanisms ranging from decoding-time constraints to prompt-level conditioning, and note that guarantees weaken as constraints become semantic rather than lexical [13]. Benchmarks of constraint following report that models satisfy fine-grained instructions inconsistently, with compliance degrading as constraints accumulate [14], and related work shows that instructions embedded in prompts can be displaced by competing content in the input [15]. A partly dissenting result finds that models are more robust to prompt phrasing than earlier work suggested, and that some reported sensitivity is an artefact of evaluation method [16]. Taken together, these studies indicate that prompt-level prohibitions are a useful but probabilistic control rather than a guarantee. A research gap emerges at the intersection of these areas. Work on hallucination generally focuses on measurement and mitigation at the model level, while work on organisational document automation generally relies on rigid templates without a generative component. Few studies report a hybrid design that fixes mandatory structural elements deterministically at the application layer and confines the model to the narrative portion, particularly for Indonesian official correspondence, which is bound by document conventions and requires multi-level approval and an audit trail. On this basis the study poses two questions. First, how can a generative-AI-assisted official correspondence drafting tool be designed so that regulation-mandated structural elements remain deterministic while the narrative portion is model-generated? Second, to what extent does that design fulfil its own design requirements when tested functionally?

3. Method

This study adopts a design science approach following the six-activity process model of Peffers et al. [9]. The approach was chosen because the primary research output is a software artifact built to address an identified practical problem rather

than a test of hypothesised relationships between variables. The process model also requires that design requirements be stated before construction and that evaluation criteria be stated before evaluation, so that fulfilment of the design can be inspected transparently. Table 1 details how the six activities were carried out.

Table 1. Execution of the design science activities

Activity	Execution in this study	Output
Problem identification	Examination of current correspondence practice drawing on the author team's professional experience, cross-referenced with selected structural provisions of PermenPANRB 21/2021 [17] and organisational letter examples.	Seven problems underlying the design requirements.
Objectives of a solution	Formulation of design requirements whose fulfilment can be inspected, together with evaluation criteria stated before construction began.	DR1 to DR7 and the evaluation criteria in Subsection 3.4.
Design and development	Iterative construction of the artifact covering data schema, template engine, language model integration, approval workflow, numbering, and print file generation.	The SCA artifact with ten data entities and seven letter types.
Demonstration	Use of the artifact on a complete letter drafting scenario, from draft creation to issuance of a numbered print file.	Generated letters with their input contexts and print files.
Evaluation	Scenario-based functional testing of DR1–DR7 fulfilment across six preserved runs, with targeted remediation and retesting.	Six test runs, eight documented defects, and cumulative functional status.
Communication	Reporting of design rationale, findings, and limitations in a scholarly article.	This article.

Source of the design requirements

The design requirements were derived from the author team's professional experience in office administration, secretarial practice, and official correspondence, then cross-referenced with selected structural provisions of PermenPANRB 21/2021 [17] and with organisational letter examples. The regulation was used as a reference standard rather than as a claim of direct legal applicability to the association. Requirements were documented before construction and mapped to inspectable functional criteria. This derivation is practitioner-informed rather than a formal requirements analysis based on surveys or structured interviews, so its representativeness is limited to the organisational context studied.

Table 2. Design requirements

Code	Identified problem	Design requirement
DR1	Letter formatting is inconsistent across drafters and over time.	Document structure is determined by templates rather than by the generative model.
DR2	Model output may contain statements unsupported by the drafter's input.	Every draft must pass human inspection and editing before submission.
DR3	Multi-level approval takes place outside the system and is hard to trace.	Approval workflow is integrated and configurable per letter.

Code	Identified problem	Design requirement
DR4	Manual numbering produces duplicates and skipped numbers.	Numbers are issued deterministically and only at final approval.
DR5	Access to letters and master data is uncontrolled.	Authority is constrained by user role.
DR6	Drafting and approval history is not recorded.	Every consequential action is logged and traceable.
DR7	Failure of the model service can halt work.	Failures are handled without corrupting data or stopping the workflow.

Core design of the artifact

The core design separates structural control from narrative generation. Each letter type has a versioned template holding the opening salutation, closing salutation, standard closing paragraph, system instruction, and reference-standard metadata. During drafting, the application sends the system instruction together with the drafter's context, recipient record, and official organisation name to the model, and receives only the narrative portion. The application then assembles the final text by joining the opening salutation, model narrative, standard closing paragraph, and closing salutation. The document number, letterhead, date, recipient block, and signature block are never model-generated; they are constructed from master and transaction data. Figure 1 summarises the resulting control boundary.

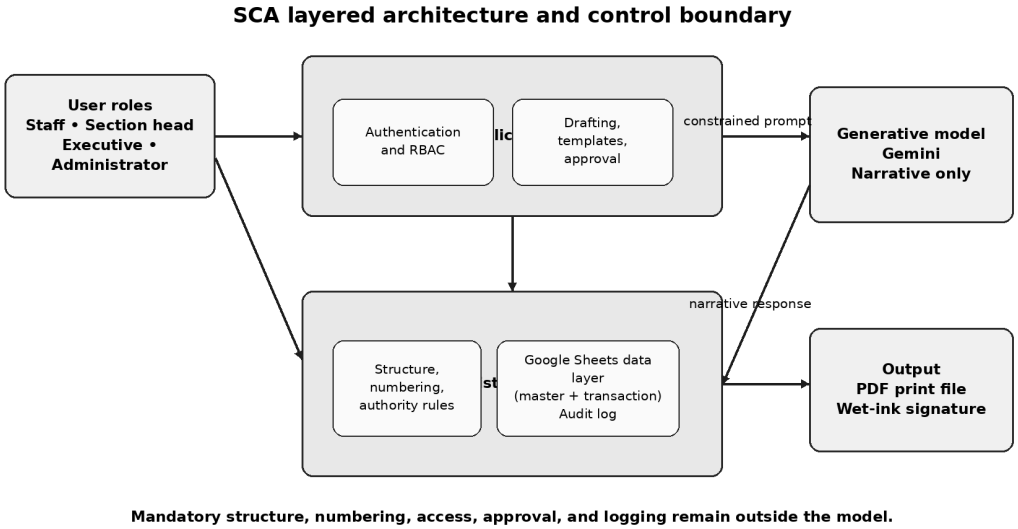

Figure 1. SCA layered architecture and control boundary

Implementation environment

The artifact was implemented as a web application in PHP without a framework, using an online spreadsheet as the data layer through its official programming interface, and the gemini-3-flash-preview model as the generation engine with a temperature of 0.3 and an output limit of 1024 tokens. Print files are produced with the mPDF library version 8.3.1. The data layer comprises ten entities: users, organisation, recipients, letter types, templates, number counters, letters, approval log, audit log, and configuration. Testing was executed in an isolated container running PHP 7.4.33 in order to approximate the deployment environment.

Evaluation criteria and procedure

The evaluation criteria were fixed before testing began. Functional conformance was defined as the extent to which artifact behaviour matched the specification for design requirements DR1–DR7. Fifty-four scenarios were distributed across nine functional groups. Each scenario specified preconditions, steps, and an expected result. During harness execution, a scenario could receive PASS, FAIL, BLOCKED, N/A, or REQUIRES_MANUAL_VERIFICATION. Eight scenarios required human judgement: LLM-01, LLM-07, LLM-08, and PDF-02 through PDF-06. The test agent did not decide these semantic or visual criteria. After the final run, the preserved evidence packages were reviewed manually by the first author against the predefined criteria and the verdicts were recorded separately. Defect severity was defined before reporting: Critical for exposure or failures that could compromise security or document validity; Major for failure of a core function or authority path; and Minor for quality or layout defects that did not independently invalidate the workflow.

Testing was carried out by the development team with the assistance of a programming agent, not by independent testers. The manual reviewer was also part of the development team; therefore, the manual verdicts are functional checks rather than independent expert validation. Testing used a separate test spreadsheet whose identity was verified as different from production before live API access. Earlier runs were preserved rather than overwritten. Each confirmed defect was documented, remediated in a separate code change, and retested. The agent retained semantic or visual scenarios for manual review, after which the reviewer recorded the final PASS verdicts using the preserved text and PDF evidence.

4. Result and Discussion

The resulting artifact

The resulting artifact is a web application named the Smart Correspondence Assistant. It comprises five modules: authentication and access control, letter drafting, narrative generation, approval and numbering, and print file generation. Four user roles are distinguished, namely staff, section head, executive, and administrator, each with a different scope of authority.

The data layer is organised into ten entities. Six are master data, namely users, organisation, recipients, letter types, templates, and configuration. Four are transactional, namely letters, number counters, approval log, and audit log. This separation also governs the caching strategy, with a longer retention period for master data and a shorter one for transactional data, in order to reduce the number of programming interface calls without sacrificing data freshness.

The approval workflow is configurable per letter through two flags, one for section-head approval and one for executive approval. The combination yields four paths: both levels, section head only, executive only, or no approval. A document number is issued only when a letter reaches approved status, using a configurable pattern and a counter scoped by letter type, unit code, and year. Figure 2 shows the complete workflow and the two points at which human judgement remains mandatory.

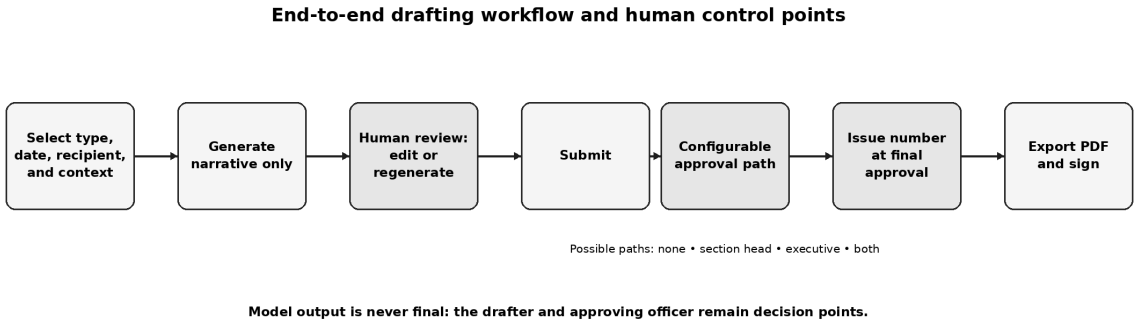

Figure 2. End-to-end drafting workflow and human control points

Demonstration

The demonstration used a request-for-audience letter scenario. The drafter selects the letter type, the letter date, and the recipient, then enters the subject and a context describing the purpose of the audience, the proposed time, and the composition of the delegation. The application sends that context together with the template system instruction and the official organisation name to the model, and receives a narrative portion of two to three paragraphs. The raw text returned by the model was confirmed to contain no letterhead, number, opening salutation, standard closing paragraph, or signature block. All of those elements are added by the application at the assembly stage.

The drafter may edit the generated narrative manually or request regeneration, and then submits the letter along the selected approval path. After final approval, the application issues the document number and produces a PDF print file containing the organisation letterhead, metadata, recipient block, letter body, and signature block. Signing is performed in wet ink after printing. Figure 3 shows a representative dashboard populated only with synthetic test fixtures.

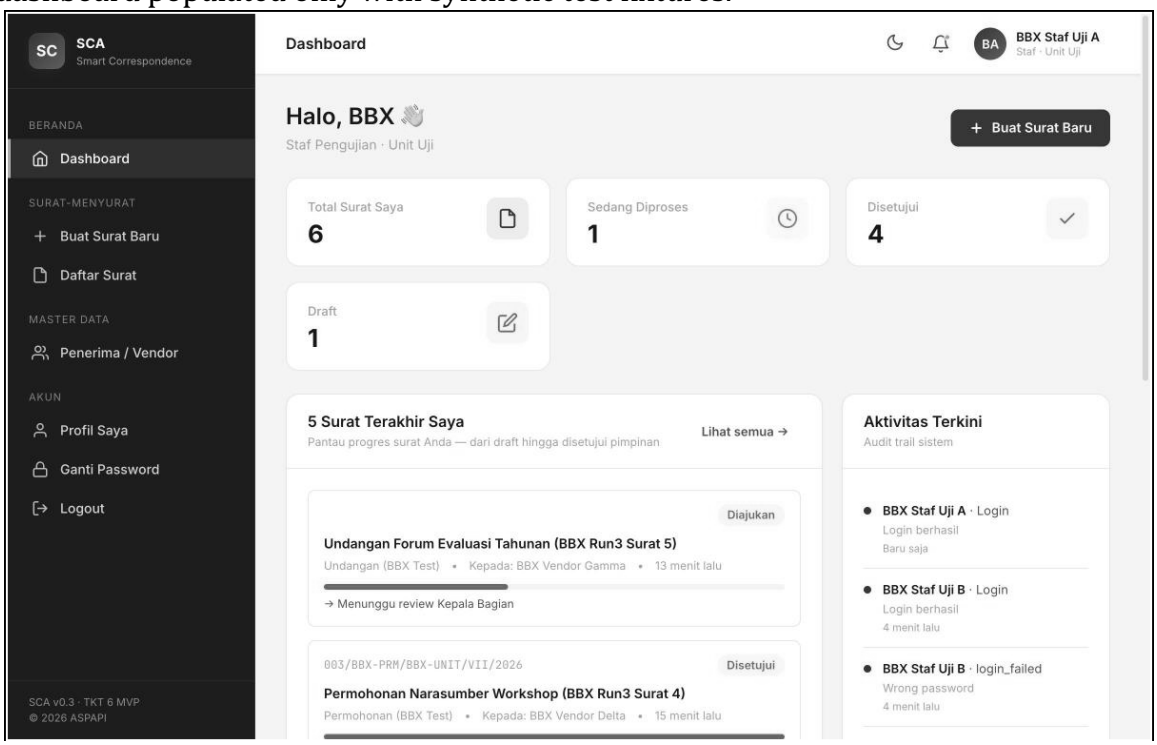

Figure 3. Representative SCA dashboard using synthetic test fixtures

Functional testing results

Fifty-four scenarios were constructed and grouped by the function under test, as presented in Table 3. Each group is mapped to the design requirements it examines.

Table 3. Test scenario groups

Code	Functional group	Count	Requirements examined
AUTH	Authentication and session	8	DR5
CRT	Letter drafting	6	DR1, DR2
LLM	Narrative generation	8	DR1, DR2, DR7
WF	Approval workflow	8	DR3
NUM	Document numbering	4	DR4
PDF	Print file generation	6	DR1
SEC	Access control	6	DR5
AUD	Audit trail	4	DR6
DTA	Data and interface	4	DR1, DR3

Testing proceeded in six runs. The first two runs were safety-gated and largely blocked because the schema and isolated test environment were not yet ready. Run 3 was the first complete execution of all scenarios and identified two failed scenarios plus additional defects observed during the same workflow. Runs 4–6 were targeted retests after remediation. Table 4 reports the status recorded at the end of each automated stage. The reduction from forty-eight recorded PASS statuses in Run 5 to forty-seven in Run 6 reflects the conservative reclassification of LLM-01 to pending manual review for five newly generated letters, not a functional regression. Manual sign-off was completed after Run 6 and is reported separately below.

Table 4. Summary of the six test runs

Run	Purpose / re-executed scope	Recorded PASS	FAIL	Pending manual	Blocked
1	Initial safety-gated run	1	1	0	52
2	SEC-06 remediation retest	2	0	0	52
3	First complete execution of all 54 scenarios	52	2	0	0
4	13 targeted retests after six remediations	48	0	6	0
5	6 targeted retests after spreadsheet-write remediation	48	0	6	0
6	3 targeted retests using five new letters	47	0	7	0

Across the six runs, eight defects were identified: two Critical, three Major, and three Minor. All eight were remediated and retested. For C8, the second-layer code safeguard was verified through nine synthetic unit tests but was not triggered by the

five live model outputs generated in Run 6. Table 5 summarises the defects and remediation.

Table 5. Identified defects and remediation

Code	Defect description	Severity	Remediation
C1	Service credential file, source code archives, and a diagnostic page were reachable over HTTP without authentication.	Critical	Files moved outside the document root and access blocked at the server layer.
C2	The letter detail page did not enforce ownership, so a staff user could open another staff user's letter by guessing the identifier.	Major	Ownership checks added to the detail page and to print file generation.
C3	The drafting form never wrote the approval requirement flags, so new letters fell through to the no-approval path.	Major	Flags written explicitly from the form; empty values treated as the strict path.
C4	The organisation name was never sent to the model, so the model produced a name variant absent from master data.	Major	Organisation name included in the prompt, with a post-generation match check.
C5	The place and date line was printed twice in the print file, drawn from two different date fields.	Minor	Second line removed; only the letter date retained.
C6	Signature block margins pushed the signature onto a new page for short letters.	Minor	Margin reduced from 45 mm to 22 mm.
C7	Data writes used a value-interpreting mode, so dates were converted to serial numbers and numbering fell back to an epoch value with a risk of duplicate sequences.	Critical	Write mode changed to raw; numbering now refuses to proceed when a date cannot be parsed.
C8	The model occasionally wrote its own closing paragraph despite an explicit prohibition, producing two closing paragraphs.	Minor	Instruction strengthened and automatic trimming added as a second layer.

Several defects provide before-and-after evidence. For C1, forty-four probes against sensitive paths initially returned successful responses; after remediation, all forty-four returned access denials while four public controls remained accessible. For C4, none of seven letters generated before remediation reproduced the organisation name exactly as stored in master data, whereas all four letters in Run 4 and all five new letters in Run 6 reproduced it exactly. For C8, none of the five new Run 6 letters contained a duplicate closing paragraph. The code-level trimming safeguard did not

trigger on those live outputs and therefore remains demonstrated only through the nine synthetic tests.

At the close of automated Run 6, forty-seven scenarios were recorded as PASS and seven remained pending manual review. One of the forty-seven recorded PASS statuses, LLM-07, had already been decided manually in an earlier development-team review; the other forty-six were decided through objective functional checks. The seven pending scenarios were LLM-01, LLM-08, and PDF-02 through PDF-06. The first author then reviewed the latest preserved evidence against the predefined criteria for semantic relevance, exact organisation-name use, letterhead, printed metadata, body completeness, signature block, and page layout. All seven received manual PASS verdicts. The final cumulative status was therefore fifty-four PASS, zero FAIL, zero BLOCKED, and zero N/A, comprising forty-six objective functional verdicts and eight manual semantic or visual verdicts. These manual checks were conducted within the development team and should not be interpreted as independent expert validation.

Fulfilment of the design requirements

Table 6 maps each design requirement to the available evidence. Statuses distinguish fulfilled, fulfilled after remediation, and partially fulfilled. Following completion of the eight manual semantic and visual checks, DR1 and DR2 were classified as fulfilled; DR6 remains partially fulfilled because audit logging is best-effort and does not abort a transaction on failure.

Table 6. Fulfilment of the design requirements

Code	Status	Evidence and notes
DR1	Fulfilled after remediation and manual verification	Template assembly and structural checks passed. PDF-02–PDF-06 were manually reviewed against the preserved evidence and received PASS verdicts.
DR2	Fulfilled after remediation and manual verification	Preview, editing, and regeneration functioned. C4 was remediated, and LLM-01, LLM-07, and LLM-08 received manual PASS verdicts after evidence review.
DR3	Fulfilled after remediation	All four workflow combinations behaved as specified after C3 was remediated.
DR4	Fulfilled after remediation	Numbers are issued only at final approval and increment by letter type, unit, and year after C7 was remediated.
DR5	Fulfilled after remediation	Role guards functioned as specified after C1 and C2 were remediated and retested.
DR6	Partially fulfilled	All tested actions were logged, but logging is best-effort and does not abort the transaction on failure.
DR7	Fulfilled	Model-service failure produced an intelligible message without persisting a partial letter record.

Discussion

The evaluation findings point to three transferable design lessons. The first concerns the choice of storage layer. An online spreadsheet enforces neither schema nor data type. Two failure modes followed, both silent. Columns absent from the header row were discarded by the adaptation layer, so features already written in code

did not operate on the data without producing any signal. Date values written under a value-interpreting mode were converted into serial numbers, so the month and year computation in the numbering routine fell back to an epoch value and could yield duplicate sequence numbers within the same letter type and unit. Both modes shift the burden of maintaining data integrity entirely onto the application. Selecting a storage layer without schema enforcement does lower the barrier for administrative users to maintain master data, but it demands explicit integrity checks at the application layer.

The second lesson concerns the locus of control. A template-grounded design controls the document at the assembly stage, not at the generation stage. That control was robust for structural elements because the letterhead, number, date, salutations, and signature block are constructed by the application. It did not automatically guarantee factual accuracy inside the narrative. Before remediation, the model consistently wrote a variant of the organisation name that appeared neither in master data nor in the drafter's input. The model also occasionally wrote its own closing paragraph despite an explicit prohibition. Both findings show that prompt-level prohibitions are probabilistic [14]. The remediation therefore uses two layers: a strengthened instruction and a code-level output check. In five live Run 6 letters the second layer was not triggered, so its live reliability remains unproven even though nine synthetic tests passed.

The third lesson concerns defect propagation. A small defect in the drafting form, namely approval requirement flags that were never written, propagated into an institutional problem. Empty flags were interpreted as the no-approval path, so letters were approved immediately upon submission and the drafter was recorded as the final approver. In the print file this presented a staff member as the signatory of an outgoing organisational letter, which is not institutionally valid. This sequence shows that in official document systems a seemingly trivial defect in an input interface can end in a document whose validity is compromised, which argues for always setting default values on authority paths to the strictest available option.

The finding regarding organisation name fabrication also supplies empirical grounding for requirement DR2. The mandatory human review in this design was originally specified as a precaution based on the hallucination literature [3], [5]. The test results show that the concern materialised in a specific form, namely an error concerning an entity that was in fact held in the application's own master data but was not included in the context sent to the model. This suggests that designers of comparable systems should explicitly audit which entities appear in generated output but were never supplied as input, because such entities originate entirely from the model's internal knowledge.

5. Conclusion

This study designed, built, and functionally evaluated a generative-AI-assisted official correspondence artifact that separates application-controlled structure from model-generated narrative. Seven design requirements were formulated from practitioner experience, organisational letter examples, and a reference standard. The artifact implements seven letter types, configurable approval, deterministic numbering, role-based access, audit logging, and PDF generation. Across six runs, eight defects were identified, remediated, and retested. After post-run manual review of the semantic and visual evidence, the final cumulative status was fifty-four of fifty-four

scenarios passing: forty-six through objective functional checks and eight through manual review within the development team.

The theoretical contribution is a transferable design pattern that places reference-standard structural elements and authority controls at the application assembly layer while confining the generative model to reviewable narrative content. Following Gregor and Hevner [10], the contribution is positioned as an instantiation with explicit design rationale rather than a complete design theory. The evaluation also shows the limit of template grounding: assembly-stage control does not guarantee factual accuracy within generated text, so supplied-entity checks and human review remain necessary. The practical contribution is a working artifact, a design rationale, and an auditable defect-remediation history for comparable correspondence systems.

Several limitations remain. First, testing was performed by the development team with programming-agent assistance rather than by independent testers. Second, the eight semantic or visual scenarios were manually verified by a member of the development team rather than by independent experts, so confirmation bias cannot be excluded; the study does not claim independent expert validation, usability, user acceptance, or population-level effectiveness. Third, the design requirements were derived within one organisational context. Fourth, the second-layer closing-paragraph safeguard passed synthetic tests but was not triggered by the five live Run 6 outputs. Fifth, the number of generated letters examined for factual and layout behaviour remains small.

Further research is directed towards three areas. The first is assessment of output quality by official-correspondence and document-convention experts not involved in development, using a scaled instrument covering structure, content, language, and readiness for use. The second is measurement of effectiveness among users through a comparison of manual and artifact-assisted drafting on an adequate sample, attending to drafting time and the number of revision cycles. The third is robustness testing of the output inspection layer on a substantially larger number of letters and a wider variation of input contexts, in order to obtain a defensible estimate of its error rate.

6. References

[1] O. R. Danar, "Digital transformation of Indonesian administration and bureaucratic system," International Journal of Electronic Governance, vol. 16, no. 2, pp. 152-171, 2024, doi: 10.1504/IJEG.2024.140789.

[2] B. Kusumasari, "Digital democracy and public administration reform in Indonesia," International Journal of Electronic Governance, vol. 10, no. 3, pp. 317-337, 2018, doi: 10.1504/IJEG.2018.095937.

[3] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," ACM Computing Surveys, vol. 55, no. 12, art. 248, pp. 1-38, 2023, doi: 10.1145/3571730.

[4] S. Nzobonimpa, J.-F. Savard, and J. Lawarée, "Generative AI in public administration: evaluating a fine-tuned large language model for policy briefing notes," Science and Public Policy, 2026, doi: 10.1093/scipol/scag026.

[5] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald, "On faithfulness and factuality in abstractive summarization," in Proc. 58th Annu. Meeting Assoc. Comput. Linguistics, 2020, pp. 1906-1919, doi: 10.18653/v1/2020.acl-main.173.

[6] N. Haug, S. Dan, and I. Mergel, "Digitally-induced change in the public sector: a systematic review and research agenda," Public Management Review, vol. 26, no. 7, pp. 1963-1987, 2024, doi: 10.1080/14719037.2023.2234917.
[7] I. Mergel, N. Edelmann, and N. Haug, "Defining digital transformation: results from expert interviews," Government Information Quarterly, vol. 36, no. 4, art. 101385, 2019, doi: 10.1016/j.giq.2019.06.002.
[8] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," MIS Quarterly, vol. 28, no. 1, pp. 75-105, 2004, doi: 10.2307/25148625.
[9] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," Journal of Management Information Systems, vol. 24, no. 3, pp. 45-77, 2007, doi: 10.2753/MIS0742-1222240302.
[10] S. Gregor and A. R. Hevner, "Positioning and presenting design science research for maximum impact," MIS Quarterly, vol. 37, no. 2, pp. 337-355, 2013, doi: 10.25300/MISQ/2013/37.2.01.
[11] J. Laux, "Institutionalised distrust and human oversight of artificial intelligence: towards a democratic design of AI governance under the European Union AI Act," AI & Society, vol. 39, no. 6, pp. 2853-2866, 2024, doi: 10.1007/s00146-023-01777-z.
[12] B. Green, "The flaws of policies requiring human oversight of government algorithms," Computer Law & Security Review, vol. 45, art. 105681, 2022, doi: 10.1016/j.clsr.2022.105681.
[13] H. Zhang, H. Song, S. Li, M. Zhou, and D. Song, "A survey of controllable text generation using transformer-based pre-trained language models," ACM Computing Surveys, vol. 56, no. 3, art. 64, pp. 1-37, 2024, doi: 10.1145/3617680.
[14] Y. Jiang, Y. Wang, X. Zeng, W. Zhong, L. Li, F. Mi, L. Shang, X. Jiang, Q. Liu, and W. Wang, "FollowBench: a multi-level fine-grained constraints following benchmark for large language models," in Proc. 62nd Annu. Meeting Assoc. Comput. Linguistics (Vol. 1: Long Papers), 2024, pp. 4667-4688, doi: 10.18653/v1/2024.acl-long.257.
[15] Z. Li, B. Peng, P. He, and X. Yan, "Evaluating the instruction-following robustness of large language models to prompt injection," in Proc. 2024 Conf. Empirical Methods Natural Language Processing, 2024, pp. 557-568, doi: 10.18653/v1/2024.emnlp-main.33.
[16] A. Hua, K. Tang, C. Gu, J. Gu, E. Wong, and Y. Qin, "Flaw or artifact? Rethinking prompt sensitivity in evaluating LLMs," in Proc. 2025 Conf. Empirical Methods Natural Language Processing, 2025, pp. 19889-19899, doi: 10.18653/v1/2025.emnlp-main.1006.
[17] Ministry of Administrative and Bureaucratic Reform of the Republic of Indonesia, "Regulation No. 21 of 2021 on Official Correspondence within the Ministry of Administrative and Bureaucratic Reform," 2021. Available: https://peraturan.bpk.go.id/Details/170616/permen-pan-rb-no-21-tahun-2021