SOFTWARE ASSURANCE / SAMPLE REPORT

Web application Technical review

Security, reliability and change readiness.

EXAMPLE
All application details and findings are fictional. No real system was reviewed.

REPORT	DETAIL
Prepared by	Antony Doyle / Siege21
Date / revision	27 September 2026 / 2.0
System	Example Workspace - fictional task and document application
Evidence	Illustrative scenarios; no code inspection or test execution

01 Executive assessment

The example identifies four issues: missing workspace access checks, repeat requests creating duplicate tasks, sensitive values in logs, and release tests that do not block deployment.

The access-control issue is the first priority because it permits one workspace to read another's documents. The remaining actions protect data integrity, reduce information exposure and prevent regressions.

Illustrative severity: 1 High / 2 Medium / 1 Low. Confidence: not assessed. Ratings demonstrate prioritisation; they are not measured results.

Immediate priorities
1. Enforce workspace membership on document access.
2. Make task creation safe to retry.
3. Remove sensitive values from logs and enforce release checks.

02 Scope, method and limitations

ASPECT	DETAIL
Coverage	Document access, task creation, application logging and release controls.
Approach	Focused code review supported by agreed test cases. Actual execution for this sample: none.
Boundaries	No penetration test, infrastructure audit, load test or compliance certification.
Evidence	S-01 to S-04 are invented examples. No repository, build identifier or runtime evidence exists for this sample.

03 System summary

Example Workspace is a fictional browser-based application for managing tasks and documents. A web interface calls an API backed by a database and file store. Users belong to separate workspaces; administrators manage membership.

The key boundary is between workspaces: signing in must not give a user access to another workspace's records. Background jobs and retries must preserve the same permissions and data rules.

04 Assessment by domain

Illustrative coverage summary. References link to details.

AREA / STATUS	SUMMARY	PRIORITY	DETAIL
Access control Issue	Document retrieval omits a workspace-membership check.	P1	E-001 / p4
Data integrity Issue	Retrying a task request can create a duplicate record.	P2	E-002 / p4
Sensitive data Issue	Error logs retain full request bodies.	P2	E-003 / p5
Release controls Issue	A release can proceed without passing automated tests.	P3	E-004 / p5
Authentication Design strength	Central sign-in is part of the example design.	Maintain	Strengths / p6
Recovery, dependencies, performance Not assessed	No evidence is available for these areas in this sample.	Unrated	Scope / p2

05 Finding register

ID	SEVERITY / TYPE	PRIORITY
E-001	High / Security	P1 - Before release
E-002	Medium / Correctness	P2 - Before affected workflow is relied on
E-003	Medium / Data handling	P2 - Before sensitive data is processed
E-004	Low / Engineering control	P3 - Next release-process improvement

Severity describes consequence; priority describes action order. All findings are illustrative, with confidence not assessed. “Not assessed” is not a pass. Effort estimates in the details are relative, not quotations.

06 Findings in detail

E-001 Documents can be read across workspaces

HIGH / SECURITY / P1 | Confidence: illustrative / not assessed

FIELD	DETAIL
Observation	The example document handler checks sign-in but not membership of the document's workspace.
Impact / conditions	A signed-in user who obtains another document's ID can read it. This crosses the intended workspace boundary; the example does not imply anonymous access.
Evidence scenario	S-01: request a Workspace B document using a Workspace A account. Example outcome: content returned. Expected: access denied.
Recommendation	Check workspace membership and document permissions before reading metadata or file contents. Apply the same rule to previews and downloads.
Validation / effort	Cross-workspace requests return no content; authorised reads still work. Include preview and download tests. Medium effort.

E-002 Request retries create duplicate tasks

MEDIUM / CORRECTNESS / P2 | Confidence: illustrative / not assessed

FIELD	DETAIL
Observation	Each task-creation request inserts a record, even when it repeats an earlier request after a timeout.
Impact / conditions	A network interruption followed by a retry can duplicate work and notifications. Separate, intentional task requests must remain valid.
Evidence scenario	S-02: submit the same operation twice. Example outcome: two task IDs. Expected: the original task is returned for a retry.
Recommendation	Assign an operation identifier and enforce uniqueness for that operation and workspace. Reuse the original result when a retry arrives.
Validation / effort	Sequential and concurrent retries create one task; distinct operations create separate tasks. Small to medium effort.

06 Findings in detail / continued

E-003 Logs retain unnecessary request content

MEDIUM / DATA HANDLING / P2 | Confidence: illustrative / not assessed

FIELD	DETAIL
Observation	The example error handler logs the complete request body, including document text and reset tokens where present.
Impact / conditions	People or services with log access may see sensitive content. A reset token is useful only while valid; ordinary document text may remain sensitive for longer.
Evidence scenario	S-03: trigger a validation error with synthetic sensitive fields. Example outcome: the full payload appears in logs.
Recommendation	Log approved diagnostic fields only, such as a request ID and error category. Redact secrets, limit access and review existing retention.
Validation / effort	Synthetic tokens and document text are absent from normal and error logs; request IDs still support diagnosis. Small effort.

E-004 Test failures do not block release

LOW / ENGINEERING CONTROL / P3 | Confidence: illustrative / not assessed

FIELD	DETAIL
Observation	Automated tests exist, but the example deployment process can run without a successful test result.
Impact / conditions	A regression can reach users despite being detectable by existing tests. This is a release-control gap, not a demonstrated vulnerability.
Evidence scenario	S-04: consider a release whose required test fails. The example workflow has no rule that prevents deployment.
Recommendation	Require the relevant tests to pass for the exact build being deployed. Include regression cases for E-001 to E-003.
Validation / effort	A deliberate failing test blocks deployment; a passing build can proceed through the agreed release process. Small effort.

07 Strengths

The example design includes central sign-in, a shared validation layer and automated tests. These provide useful foundations, but do not establish that every operation enforces permissions or that tests run before release.

Strengths are part of the fictional scenario. In an actual review, each would be tied to inspected evidence. Untested areas remain unassessed.

08 Remediation sequence

ORDER	ACTION	EXIT EVIDENCE
1 / E-001	Enforce document access boundaries.	Cross-workspace reads, previews and downloads denied.
2 / E-002	Make retried task creation safe.	Retry and concurrency cases create one task.
3 / E-003	Minimise and protect logs.	Sensitive test values absent; diagnostics remain useful.
4 / E-004	Require tests before deployment.	Failed checks block the exact release build.

Proposed owner for each action: application engineering. Owners and dates require agreement. E-004 protects the earlier fixes; its regression suite depends on those acceptance cases.

09 Open questions and accepted risks

Confirm whether shared links intentionally allow access outside a workspace, which integrations retry task creation, and who can read retained logs. These answers may change scope and priority. No risk acceptance is recorded.

10 Verification record

RECORD	STATUS
S-01 to S-04	Illustrative scenarios only; not executed.
Build / test evidence	Not applicable to this sample. A completed review records the exact version, method, result and evidence reference.
Retest	Not performed. Each fix would be recorded as resolved, partially resolved, not resolved or cannot verify.

Assurance boundary

This is an example of report structure and level of detail, not an assessment of a real application. Actual findings depend on the agreed scope and evidence. A focused review cannot guarantee the absence of other defects.