

Sanitised QA / Bug Reporting Portfolio Sample

Application Support / QA Testing / Web Support / Issue Investigation

Purpose: This document is a sanitised portfolio sample designed to demonstrate clear bug reporting, QA thinking, troubleshooting, and software issue documentation for junior Application Support, Product Support, QA/Test Analyst, Web Support, and Technical Support Analyst roles.

Source context: The reports are rewritten from real bug reports prepared during a web development internship in a staging/testing environment.

System context: These reports were based on testing a subscription-based rental/e-commerce platform with customer-facing product pages, subscription-based product availability, admin rental workflows, product filtering, and issue-tracking processes. The reports focus on QA documentation, workflow testing, browser compatibility, and business logic validation.

Sanitisation note: Company names, staging/admin URLs, exact page paths, internal account names, personal names, email addresses, and private contact details have been removed or generalised. The examples below are intended to show reporting structure and investigation approach, not to expose a private system.

Skills Demonstrated

- Writing clear summaries, impacts, steps to reproduce, expected results, and actual results.
- Testing business logic, workflow state, browser compatibility, and data validation issues.
- Checking reproducibility and adding useful investigation notes for developers/support teams.
- Communicating technical problems in a structured way suitable for tickets, QA reports, and support escalation.

Selected Report Summary

Report	Issue Area	Severity	Why it matters
01	Rental limit logic	Major	Blocks valid customer rentals and suggests stale workflow state.
02	Safari dashboard layout	Minor	Shows browser compatibility and UI regression testing.
03	Combined filters 500 error	Major	Breaks admin filtering and points to back-end query handling.
04	Missing RRP availability logic	Major	Mislabels a data issue as a subscription restriction.
05	Unresolved issue state consistency	Major	Allows problem items to appear rentable again.

Report 01 - Rental limit does not reset after completed rental

Area	Back-end / business logic
Priority	High
Severity	Major
Reproducibility	Always

Summary: A premium-tier customer account was blocked from renting another item even though no active rentals were visible.

Context: During rental workflow testing, a completed rental appeared to remain counted against the account limit. The account showed no active rentals in the customer view or admin view, but the system still displayed a limit-reached message.

Impact: Legitimate customers could be prevented from placing new rentals, creating a checkout/support issue and indicating a possible state-reset or data-consistency problem.

Steps to Reproduce

1. Log in as a premium-tier test customer.
2. Complete a rental workflow and return/finalise the item.
3. Confirm there are no active rentals visible in both customer-facing and admin-facing views.
4. Attempt to add another item to the cart.
5. Observe whether the system allows the new rental or incorrectly blocks the action.

Expected Result: A customer should be able to rent another item if they have not exceeded their active rental limit.

Actual Result: The system displayed a limit-reached message and blocked the rental, despite no active rentals being visible.

Additional Testing Notes: The issue was reproduced with a second anonymised test account created specifically to verify the rental-limit workflow.

Possible Cause / Investigation Notes: The active rental count may not be clearing correctly after return/finalisation, or the limit check may be reading stale workflow status data.

Report 02 - Safari dashboard layout shows extra scroll space and distortion

Area	Front-end / browser compatibility
Priority	Normal
Severity	Minor
Reproducibility	Observed in Safari

Summary: The admin dashboard displayed extra empty scroll space and layout distortion when viewed in Safari, especially after zooming.

Context: While checking browser behaviour, the dashboard showed an unexpected blank scrollable area at the bottom in Safari. Zooming in or out then caused top-row dashboard elements to become misaligned.

Impact: The issue reduces confidence in the admin interface and may make dashboard information harder to read for Safari users.

Steps to Reproduce

1. Open the admin dashboard in Safari.
2. At the default zoom level, scroll to the bottom of the page and observe the extra blank space.
3. Zoom in on the page and observe the dashboard row alignment.
4. Zoom out again and observe whether the distortion remains.
5. Compare the same dashboard behaviour in another browser, such as Chrome.

Expected Result: The dashboard should not show unnecessary blank scroll space, and layout elements should remain stable when the page is zoomed.

Actual Result: Safari displayed extra empty space at the bottom, and dashboard components became distorted after zooming.

Additional Testing Notes: The issue appeared browser-specific. It was not reproduced in Chrome during the same check.

Possible Cause / Investigation Notes: The cause may relate to Safari-specific CSS handling, viewport height calculations, flex/grid behaviour, or chart container sizing during zoom changes.

Report 03 - Combined Status and Brand filters trigger 500 error

Area	Back-end / filtering and query logic
Priority	High
Severity	Major
Reproducibility	Always

Summary: Applying Status and Brand filters together on the admin product list caused a server error.

Context: Other filter combinations worked normally, but combining Status and Brand filters consistently returned a 500 error. After the error occurred, the page also became noticeably laggy.

Impact: Admin users could be blocked from filtering product records, and the server error suggests a back-end query or request-handling problem that may affect reliability.

Steps to Reproduce

1. Open the admin product list.
2. Apply a Status filter using any available status option.
3. Apply a Brand filter. The order of applying the filters should not matter.
4. Observe the page response after both filters are active.
5. Note whether a 500 error appears and whether the browser tab becomes laggy.

Expected Result: Status and Brand filters should work together without causing server errors or significant page lag.

Actual Result: Combining the two filters caused a 500 server error and degraded page responsiveness.

Additional Testing Notes: The issue was reproducible regardless of whether the Status filter or Brand filter was applied first.

Possible Cause / Investigation Notes: A likely investigation area is the filter query construction, especially ambiguous column names such as status across joined tables, missing qualification, or unsafe combination of filter parameters.

Report 04 - Missing RRP causes incorrect subscription availability

Area	Back-end / data validation and business logic
Priority	Normal
Severity	Major
Reproducibility	Always

Summary: A product with a missing RRP value was incorrectly shown as unavailable, even for a customer on the highest subscription tier.

Context: While checking product availability rules, a product without a recommended retail price (RRP) value displayed a subscription restriction message. The message suggested the customer tier was the problem, but the underlying issue appeared to be incomplete product data.

Impact: Products could be incorrectly blocked from rental, customers may receive misleading messages, and support staff may investigate the wrong cause if missing product data is presented as a subscription issue.

Steps to Reproduce

1. Log in as a customer on the highest subscription tier.
2. Open a product that does not have an RRP value set.
3. Observe the product availability message shown on the page.
4. Check whether the product data is missing the RRP value.
5. Compare the expected subscription access with the actual restriction shown to the customer.

Expected Result: Missing RRP values should be handled clearly, either by using fallback logic or by flagging the product as incomplete data rather than as a subscription restriction.

Actual Result: The product was marked unavailable for the customer, and the message incorrectly implied the subscription tier was insufficient.

Additional Testing Notes: The behaviour suggested the availability check depended on the RRP field without properly handling null or empty values.

Possible Cause / Investigation Notes: The product availability logic may require stronger validation for required pricing fields and clearer error handling when product data is incomplete.

Report 05 - Unresolved item issues are not reflected in rental availability

Area	Back-end / workflow state consistency
Priority	High
Severity	Major
Reproducibility	Always

Summary: Items with unresolved issue records could lose their visible problem status and become available for rental again.

Context: During rental workflow testing, an item with an unresolved issue such as damaged or uncleaned still appeared available later in the workflow. The issue remained visible in the issue-tracking area, but the item itself no longer clearly showed the problem status in relevant rental views.

Impact: This creates a high operational risk: damaged or uncleaned items may be rented out again, admins may believe the item is clear, and issue-tracking data may become inconsistent with rental availability.

Steps to Reproduce

1. Complete the initial rental and delivery workflow for an item.
2. Process the item through return handling.
3. At the inspection stage, add a problem status such as damaged or uncleaned.
4. Continue through the workflow without resolving the issue.
5. Check whether the item becomes available again on the customer-facing side.
6. Compare the item/rental status with the unresolved issue record in the issue-tracking area.

Expected Result: Items with unresolved issues should retain their problem status, be blocked from being rented again, and show consistent issue information across rental, product, and admin views.

Actual Result: The item's visible problem status disappeared from relevant workflow views while the issue record remained unresolved, allowing the item to appear rentable again.

Additional Testing Notes: The issue demonstrated a disconnect between the issue-tracking area and the rental/item availability state.

Possible Cause / Investigation Notes: The rental workflow may be finalising or resetting item state without checking unresolved issue records, or the item availability logic may not be synchronised with the issue-tracking system.

Note: This sample is intentionally sanitised and does not include screenshots, private URLs, internal bug IDs, or client-specific identifiers. In a live QA/ticketing environment, screenshots, environment details, browser/device details, and exact version numbers would normally be attached to the relevant ticket.