

CASE STUDY

XYZ Bank HRMS — Leading a Capstone QA Cycle

Manual functional testing · Team leadership · Risk-based defect reporting

Overview

A capstone project testing an HR Staff Management System built on OrangeHRM, framed for a fictional client, XYZ Bank Limited. I led a three-person QA team through execution of 40 test cases across Login, Dashboard, PIM, My Info, Performance, and Recruitment, then authored the final report presented to project stakeholders.

My Role

Team Lead, working alongside two other testers (Salem and Adetoun). Beyond executing my own assigned modules, I coordinated module ownership across the team, reviewed each tester's defect write-ups for consistency and severity accuracy, and consolidated everything into a single stakeholder-facing report — including the risk assessment and go/no-go recommendation.

Results at a Glance

Metric	Result
Test cases executed	40
Pass rate	82.5% (33 passed)
Defects logged	7 — all Open
High-severity defects	4
Exit criteria met	No

The Finding That Mattered Most

Four separate-looking defects — two in My Info, two in PIM — turned out to share one root cause: name and date-of-birth fields across multiple modules accept unvalidated numeric or special-character input, with no error shown. In two cases (PIM), the system went further and displayed a false “Successfully Saved” confirmation, meaning a user would have no reason to suspect anything was wrong.

Rather than write these up as four independent tickets, I reported them as a single systemic gap in the risk assessment section. That framing changed the recommendation given to the team: fix the validation rule once, centrally, instead of patching each module in isolation — which matters both for engineering effort and for making sure the next module built doesn't reintroduce the same bug.

Why It Mattered — Reading the Business Context

The brief specified a banking client, so I treated the system's data as if it fed payroll, compliance reporting, and audit processes — because in a real deployment, it would. On that basis:

- Bad date-of-birth data can misstate retirement or benefits eligibility.
- Malformed names create mismatches against KYC/ID documentation held elsewhere in the bank's systems — exactly the kind of discrepancy an HR audit flags.
- A false “Successfully Saved” message is worse than a rejected save, because it removes the user's chance to notice and correct the error at the point of entry.

This is the section of the report I'm proudest of: it's not just “here's a bug,” it's “here's why this bug is expensive for this specific kind of organization.”

What I'd Do Differently Next Time

- Push for a shared, centralized validation utility to be tested once rather than per-module — I flagged this as a recommendation but didn't have the access to verify it against the actual codebase.
- Expand coverage to the 8 modules left untested this cycle (Admin, Leave, Time, Directory, Maintenance, Claim, Buzz, User Profile) before treating this as a full quality assessment rather than a snapshot.
- Introduce a lightweight peer-review checklist for defect write-ups earlier in the cycle, rather than catching consistency issues during my own final consolidation pass.

Skills Demonstrated

- Manual functional testing and structured test execution
- Team leadership: module coordination, defect-review, report consolidation
- Root-cause pattern recognition across modules, not just individual defect logging
- Risk-based severity assessment tied to a specific business/regulatory context
- Stakeholder-facing reporting and release-readiness recommendations

Full report: QA Test Summary & Bug Report (PDF) — available in the project repository.