Swift Pay QA Review Template

Module Lab: Quality Review for "Swift Pay"

Learner Name:	Noah Jamal Nabila
Date:	February 26, 2026
Feature Reviewed:	FS-P2M-v1.0: "Scan to Pay" (QR)

Part 1: Test Cases

Based on the Feature Specification (FS-P2M-v1.0) and the UI Mockup Description, the following two test cases have been prepared.

Test Case ID	Test Case Type	Test Scenario (Summary)	Test Steps (Numbered)	Expected Result
TC-01	Positive (Happy Path)	Verify successful payment to a merchant after scanning a valid QR code and entering a valid amount and correct PIN.	1. Log in to Swift Pay and navigate to the Home Screen. 2. Tap the "Scan" icon at the bottom centre of the Home Screen. 3. Point the phone camera at a valid merchant QR code (e.g., Kofi's Cafe). 4. Verify the Payment Confirmation screen loads with the merchant's name displayed at the top. 5. Enter a valid payment amount (e.g., GHS 50.00) in the Amount field. 6. Tap the "Confirm Payment" button. 7. Enter the correct 4-digit transaction PIN on the PIN Entry screen. 8. Observe the transaction result screen.	The camera opens in scan mode (AC-01). The QR code is decoded, and the Payment Confirmation screen is displayed with the merchant's name at the top (AC-02, AC-03). The "Confirm Payment" button becomes active after the amount is entered (AC-04). The PIN entry screen appears (AC-05). The transaction is processed and a "Payment Successful" message is displayed showing the amount (GHS 50.00) and merchant name (Kofi's Cafe) (AC-06, AC-07).
TC-02	Negative / Edge Case	Verify that entering an incorrect PIN three consecutive	1. Log in to Swift Pay and navigate to the Home Screen. 2. Tap the "Scan" icon and scan a valid merchant QR code.	After each incorrect PIN attempt, an error message is displayed, and the user is returned to the Payment

		times locks the transaction and displays an appropriate error.	3. Enter a valid payment amount (e.g., GHS 25.00) and tap "Confirm Payment". 4. On the PIN Entry screen, enter an incorrect 4-digit PIN. 5. Observe the error message and note that the user is returned to the Confirmation screen (per AC-08). 6. Repeat steps 3–5 two more times with incorrect PINs (3 total failed attempts).	Confirmation screen (AC-08). After 3 consecutive failed attempts, the transaction should be blocked and the user should receive a security warning message (e.g., "Too many failed attempts. Transaction locked."). Note: The spec (AC-08) does not define a retry limit, which is a gap; best practice for fintech apps is to enforce a maximum of 3 PIN attempts to prevent brute-force attacks.

Part 2: Bug Report

The following bug was identified by comparing the Feature Specification with the UI Mockup.

Bug ID:	BUG-01
Bug Title:	UI Mockup allows editing of pre-filled amount field, contradicting AC-09 (embedded QR amount should be read-only)
Severity:	☐ Critical ☑ **High** ☐ Medium ☐ Low
Priority:	☑ **High** ☐ Medium ☐ Low
Reference:	UI Mockup (Payment Confirmation Screen) vs. Feature Specification AC-09
Description / Steps to Reproduce:	The Feature Specification (AC-09) states: "If the QR code already has an amount embedded, the user should not be able to edit the amount." However, the UI Mockup for the Payment Confirmation Screen shows the Amount (GHS) input field as always editable with no visual indication that it could be disabled or made read-only. There is no design state for a locked/pre-filled amount field. **Steps:** 1. Review Feature Specification AC-09. 2. Review the UI Mockup for the Payment Confirmation Screen. 3. Compare the two: the mockup does not account for a disabled/read-only state of the amount field when the QR code contains an embedded amount.
Actual Result:	The UI Mockup shows the amount input field as always editable (active, with a numeric keypad). No disabled or read-only state is provided for scenarios where the merchant QR code contains a pre-embedded amount. If implemented as designed, users could alter a merchant-specified amount, leading to underpayment or overpayment.
Expected Result:	When a QR code with an embedded amount is scanned, the amount field on the Payment Confirmation Screen should be pre-filled and visually disabled (greyed out or read-only), preventing user modification. The UI Mockup should include a distinct design state for this scenario, consistent with AC-09.

Part 3: Process Improvement Suggestion

As a data specialist, I would recommend implementing a specification-to-mockup traceability matrix during the design phase. Before UI mockups are finalized, a data specialist should map every acceptance criterion to its corresponding UI state, ensuring each data condition (e.g., "amount embedded in QR" vs. "no amount embedded") has a defined visual representation. This structured, data-driven cross-referencing approach would have immediately flagged that AC-09’s read-only amount condition had no matching disabled-field state in the mockup, preventing this gap from reaching the QA review stage.