QA Final Report: Choithrams E-Commerce Platform

Target Application: Choithrams E-commerce Platform (Web & Mobile)
Project Goal: Establish a robust, reusable QA framework and conduct a cross-platform quality audit.

QA Lead: Nadia Ghulam Ali
Date: October 3, 2025

Software Testing Life Cycle (STLC) Plan:

Target Application: Choithrams E-commerce (Web Portal and Mobile App)
URL = https://choithramsgcc.com/en

Objective: To systematically ensure the quality, reliability, and performance of the Choithrams e-commerce platform across all user access points.

1. Requirement Analysis

Activity	Description	Cross-Platform Focus
Document Review	Analyze Business Requirements Documents (BRD) and Functional Specifications (FSD) for the **Checkout ,Pricing Logic, and Inventory Management.**	Prioritize requirements that impact **data synchronization** (e.g., cart updates, order status) between Web and Mobile.
Scope Definition	Clearly define **in-scope** areas (e.g., Payments, Cart, Product Search) and **out-of-scope** areas (e.g., backend warehouse management system).	Ensure scope covers both Web UI responsiveness and native Mobile functionality (e.g., Apple Pay integration, push notifications).

Acceptance Criteria (AC) Definition	Define clear, measurable conditions for success (e.g., "Tax must be calculated on final total").	Define AC for mobile gestures and responsiveness on different screen sizes/orientations.

2. Test Planning

Activity	Description	Cross-Platform Focus
Test Strategy Development	Define the levels of testing (Unit, Integration, System, Acceptance) and types (Functional, Non-Functional, Security).	**Prioritize Cross-Platform Synchronization Testing** and Mobile-specific testing (e.g., network switching, device interrupts).
Resource Allocation	Assign QA leads and establish timelines based on the complexity of the e-commerce platform's core modules (Pricing/Payment).	Ensure time is allocated for testing on **mobile devices (iOS)** and multiple browsers (Chrome, Safari).
Entry/Exit Criteria	Define criteria for starting the testing phase (e.g., "Smoke test passes on the Staging environment") and exiting (e.g., "All P1/P2 bugs resolved").	Mandate that all P1 bugs must be verified on **both Web and Mobile** platforms before release.

3. Test Case Development

Activity	Description	Cross-Platform Focus

Test Case Creation	Create detailed, standardized test cases covering core functionality and high-risk areas.	**Must include Negative Test Cases** targeting Inventory failures (Item A in cart, stock drops to zero) and Security flaws (SQL injection on login/search).
Test Data Generation	Create realistic test accounts and test data scenarios (e.g., multiple addresses, saved payment methods, loyalty program data).	Generate test data for **concurrent actions** (e.g., two users adding the last item to the cart simultaneously) to test server concurrency.
Review and Approval	Peer review all test cases within the QA team, followed by sign-off from the Product Owner (PO).	Verify that test cases explicitly call out the platform (Web/Mobile) where the action should be executed.

4. Test Environment Setup

Activity	Description	Cross-Platform Focus
Environment Preparation	Set up dedicated Staging and Pre-Production environments that mirror the Production database and configurations.	**Web:** Provision necessary browser versions (Chrome, Firefox, Safari) and screen resolutions. **Mobile:** Provision physical devices/emulators for latest iOS and Android versions.
Test Tool Setup	Install necessary tools for **Performance/Load Testing** (e.g., checking checkout speed under high load) and **Security Scanning**.	Ensure Mobile Test Automation frameworks (e.g., Appium) are correctly configured to interact with native elements.

Sanity Check	Execute the basic **Smoke Test Suite** to ensure the environment is stable and ready for full execution.	Verify all integrations (Payment Gateway, Tax Service, Shipping Calculator) are live and functioning in the test environment.

5. Test Execution

Activity	Description	Cross-Platform Focus
Execution Tracking	Systematically execute test cases and log the Pass/Fail status in the Test Management Tool Jira	**Prioritize Parallel Execution:** Run Payment Gateways tests on Web while running Push Notification tests on Mobile.
Defect Logging	Log all defects found using the **Standardized Bug Report Template (BRT)**, ensuring clear steps to reproduce and classification (P1, P2, P3).	**Mandatory Cross-Verification:** Every defect found on one platform (e.g., Web) must be immediately cross-verified on the other platform (Mobile) to check for a pervasive bug.
Regression Testing	Re-run affected test cases after any bug fix to ensure no new defects have been introduced.	Focus regression efforts on areas surrounding the **Cart/Checkout Summary** module, as it touches almost every system component.

6. Test Cycle Closure

Activity	Description	Cross-Platform Focus

Test Summary Report	Generate a final report detailing test coverage, total defects found (by severity), and the overall quality assessment.	**Highlight Cross-Platform Defects:** Specifically call out defects that failed synchronization or rendered inconsistently across Web/Mobile.
Release Recommendation	Based on the Exit Criteria, provide a formal Go/No-Go recommendation to the stakeholders (Product Owner, Release Manager).	If any P1/P2 mobile-specific bugs remain, advise caution or blocking the mobile app update until resolution.
Artifact Maintenance	Archive all test cases, test data, and final reports. Update the reusable **QA Process and Test Case Suite** based on lessons learned.	Ensure all test scripts are updated to reflect any UI/API changes made during the release cycle.

TEST CASE DESIGN

The following Test Cases (TCs) are designed to validate the high-risk areas of the Choithrams E-commerce platform, focusing primarily on the Checkout Funnel, Synchronization, and Financial Integrity.

TC ID	Test Type	Scenario Title	Pre-conditions	Test Steps	Key Expected Result

CH-TC-001	**Synchronization**	Verify Cart Item Sync Across Web and Mobile	User must be logged in on both Web (Browser) and Mobile (Native App). The cart must be empty on both platforms.	1. On the **Web Browser**, add 3 unique items to the cart. 2. Immediately open the **Mobile App** and navigate to the Cart screen (do not refresh). 3. On the **Mobile App**, change the quantity of Item 1 from 1 to 2. 4. Return to the **Web Browser** and check the quantity of Item 1.	1. The Mobile App's cart badge must update to '3' within 5 seconds. 2. The item list on the Mobile App must show all 3 items with correct details. 3. The quantity of Item 1 on the Web Browser must immediately reflect the change to '2' without requiring a manual page refresh.

CH-TC-002	**Functional/Pricing**	Validate Discount & Tax Recalculation on Quantity Change	A Multi-Buy promotion (e.g., 20% off when buying 5 items) is active on Product X. Tax/VAT is applied to the order subtotal.	1. Add 4 units of **Product X** to the cart. Verify no discount is applied. 2. Proceed to the **Checkout Summary** page (where tax/shipping is displayed). 3. Return to the cart and change the quantity of **Product X** from 4 to 5 (triggering the discount). 4. Return to the **Checkout Summary** page.	1. The discount (20%) must be correctly applied to the subtotal on the Checkout Summary page. 2. The **VAT/Tax amount** must be correctly recalculated based on the *new, discounted subtotal*. 3. The Grand Total must accurately reflect (Subtotal - Discount + Tax).

CH-TC-003	**Negative/Inventory**	Verify System Handles Out-of-Stock Item Gracefully During Checkout	Item A is added to the cart (stock = 1). A simulated external inventory update reduces the stock of **Item A** to 0.	1. Add **Item A** to the cart. 2. Proceed to the Shipping/Address screen. 3. **(Simulate)** The backend inventory updates the stock of Item A to zero. 4. Click **"Proceed to Payment"** or **"Continue Checkout"**.	1. The user must be **prevented** from proceeding to the payment screen. 2. An immediate, clear, and non-intrusive error message must appear (e.g., "Item A is now out of stock and has been removed from your cart"). 3. The Grand Total must be **automatically reduced**, reflecting the removal of Item A.

CH-TC-004	**Boundary/Input**	Validate Shipping Address Input with Max Length and Special Characters	User must be logged in and on the Shipping Address input screen.	1. In **"Address Line 1,"** paste a string of 255 mixed characters (letters, numbers, Unicode symbols: 🌐 € # &). 2. Attempt to save the address and proceed to the Payment screen. 3. Return to the saved addresses and view the recently saved address.	1. The system must successfully **truncate or accept** the input without throwing a server error or crashing the UI. 2. The saved address must display the text correctly, without breaking the layout of the "Saved Addresses" list. 3. The payment gateway must accept the complex address string when the final order is placed.

CH-TC-005	**Regression/Payment**	Verify Saved Payment Method Selection Persists Across Session	User has a credit card or cash on delivery option selected,saved to their account.	1. Select the saved payment method (e.g., VISA ending in 1234) on the payment screen. **DO NOT SUBMIT THE ORDER.** 2. Close the browser window entirely (or kill the mobile app). 3. Re-open the site, add an item to the cart, and proceed to the Payment screen.	1. The user must be taken back to the correct step in the checkout process (not reset to step 1). 2. The previously selected payment method (e.g., VISA 1234) must remain **pre-selected** or clearly visible as the default option.

DEFECT LOG AND BUG REPORTING

This section will contain the high-impact defects discovered during the exploratory testing.

Bug Report #1 (P1 - Critical):

BR ID	CHO-B-001
Title	**Critical Cross-Platform Validation Failure: Malformed Address Input Causes Crash on Mobile, but Fulfillment Corruption on Web**
Severity	**P1 - Critical**
Priority	P1 (Immediate Hotfix for both platforms)
Environment	**Web** (Chrome v141) and **Mobile App** (iOS)
Steps to Reproduce	1. Log in and add an item to the cart. 2. Proceed to the Shipping Address screen. 3. Enter the following string into the primary address field (Address Line 1): `ddddddfdddcx$$$$$&@&)(;::3/-,($$`. 4. Confirm the address and proceed to the Delivery Slot Selection screen. 5. Choose any available slot. 6. **Complete the payment process successfully (Web) or attempt to proceed (Mobile).**
Expected Result	The system must prevent the user from saving the malformed address with an immediate validation error: "Invalid characters in address. Only letters, numbers, and standard punctuation are allowed."

Actual Result	**Web Platform Failure (Fulfillment Risk):** The system accepts the malformed address, processes the order successfully, and records the corrupted string as the final delivery address, making the order **undeliverable**. **Mobile Platform Failure (Blocking Crash):** The app accepts the malformed address but immediately **crashes** upon slot selection, resulting in session loss and cart abandonment.
Attachment	Bug_Attachment_Link

Bug Report #2 (P1 - Critical):

Field	Content
BR ID	CHO-B-002
Title	**Critical Blockage: Delivery Slot Time Options Missing on Mobile, Resulting in Validation Loop**
Severity	P1 - Critical
Priority	P1 (Immediate Hotfix)
Environment	Choithrams Mobile App (iOS/Android)
Steps to Reproduce	1. Log in, add items to the cart, and proceed to the Delivery Slot Selection screen. 2. Observe the displayed slots. Note that only the delivery dates

	(days) are visible. 3. Select one of the available dates. 4. Attempt to proceed to the next step (e.g., Payment).
Expected Result	After selecting a date, a list of available time slots (e.g., 10:00 AM - 12:00 PM) should populate. Selecting both a date and a time should allow the user to proceed.
Actual Result	No time slots are displayed, making a time selection impossible. The system returns a blocking error: "Please select time slot," which traps the user and prevents the order from being placed.
Attachment	Bug_Attachment_Link

Bug Report #3 (P2 - Major):

Field	**Content**
BR ID	CHO-B-003
Title	**Critical Data Inconsistency: Cart Quantity Reduction to 41 Causes "No Items" Message**
Severity	**P2 - Major**
Priority	P2 - (High)
Environment	Choithrams Mobile App (iOS)

Steps to Reproduce	1. Add any single product to the cart. 2. On the cart screen or product widget, tap the '+' button **51 times** to set the quantity to 51. 3. Tap the '-' button **10 times** to reduce the quantity from 51 down to 41. 4. Observe the main cart item list.
Expected Result	The cart counter should display “41”. The main cart view should display the product row with the quantity correctly set to 41, and the subtotal should reflect the cost of 41 units.
Actual Result	The cart counter/badge correctly shows '41'' items, but the main viewing area of the cart displays the message: "**No items, seems like you haven't added any products to basket yet.**" The user is blocked from checkout.
Attachment	Bug_Attachment_Link

TEST SUMMARY REPORT:

Cycle: Choithrams Regression Test Period: Q4 2025 Validation

1. EXECUTIVE SUMMARY & QUALITY ASSESSMENT

This test cycle focused on the high-risk Checkout Funnel (Cart, Shipping, and Payment modules) across both the Web Portal and the Native Mobile App.

The cycle discovered two P1 Critical (Blocking) defects that prevent users from placing orders on the Mobile App, and one P2 Major defect exposing a critical fulfillment risk on the Web Portal.

Conclusion: The application currently fails the Release Exit Criteria due to the presence of multiple P1 Critical, revenue-blocking defects.

Recommendation: NO-GO for Production Release. Immediate hotfixes are required for the defects listed below.

2. TEST EXECUTION STATUS

Total Test Cases in Scope: 5 Total Test Cases Executed: 5 Test Execution Pass Rate: 40% (2/5 TCs Passed)

Status	Count	Affected Test Cases (TCs)
PASS	**2**	CH-TC-003 (Negative/Inventory), CH-TC-005 (Regression/Payment)
FAIL	**3**	CH-TC-001, CH-TC-002, CH-TC-004

3. DEFECT ANALYSIS AND SEVERITY BREAKDOWN

A total of 3 defects were logged and verified using the Standardized Bug Report

Severity	Count	Impact	Status
P1 - Critical	**2**	Directly blocks revenue and prevents order placement. Requires immediate attention and hotfix.	**OPEN**
P2 - Major	**1**	Data Validation Error; causes cart display failure and blocks user from checkout.	**OPEN**

4. RECOMMENDATION

Given that the Mobile App is currently non-functional at two critical points (Cart view and Delivery Slot selection), the following actions are required before the code can be approved for production:

1. **Mandatory Hotfix:** Address CHO-B-001 and CHO-B-002 immediately.
2. **QA Verification**: Once fixes are deployed to Staging, QA must prioritize regression testing for the entire Checkout Funnel.
3. **Future Action**: Address the P2 Major fulfillment risk (CHO-B-003) in the immediate subsequent sprint.