CleanCity — Waste Pickup Scheduler

QA Test Report (Step 1: Prepared Report)

Prepared By: Meseret Akalu
Reviewed By: Mercy Benu, Viron Ochieng
Prepared Date: 2025-11-18
Project Title: CleanCity Waste Pickup Scheduler
Team / Institution: CleanCity QA Team — Software Testing Mastery in Scrum Course

---

Table of Contents

Executive Summary (page 3)

- Scope: Full functional and non-functional QA of CleanCity Waste Pickup Scheduler including core flows (auth, scheduling, admin), new features, and integrations.
- Overall status: The QA effort executed the complete planned suite. Key result: of 34 planned test cases executed, 24 passed and 10 failed. (See Test Execution Summary for details and raw outputs.)
- Pass rate (based on executed cases): 24/34 → 70.6%.
- Key findings:
    - Critical issues found: session timeout not implemented (security risk), scheduling allowed for past dates (business/ops risk).

    - High-impact admin and filter defects block a subset of workflows (admin edit, filter combination).
    - Wishlist synchronization shows intermittent loss — requires a hotfix (SHOP-4589 referenced).
- Recommendation: Proceed with a controlled, phased release (10% initial ramp) only after addressing at minimum the critical security and scheduling fixes or ensuring compensating monitoring and rollback capabilities. Implement enhanced monitoring on payment gateway and login-related flows for first 72 hours.

---

3. Test Strategy & Approach (page 4)

Objectives

- Validate that business-critical journeys are correct, secure, performant, and accessible.
- Reduce release risk via prioritized testing and automation.

Scope (Functional Areas)

- User Authentication & Account Management (registration, login/logout, password reset, profile, preferences)
- Scheduling & Pickup flows (create, edit, cancel pickups, validation including date/time)
- Product / Service Catalog & Search (browse, filters, advanced search)
- Cart, Checkout & Payments (including 5 new payment methods)
- Admin & Back-office workflows (edit requests, statistics)
- New features: personalized recommendations, wishlist sharing, AR visualization, in-app chat

Non-Functional Areas

- Performance/load under realistic and peak conditions
- Compatibility across Android (v10–v14), iOS (v15–v18), major browsers
- Security: input validation, session management, secure storage
- Accessibility: WCAG conformance, screen-reader tests

Test Design & Techniques

- Risk-based prioritization (payments, auth, scheduling first)
- Test types used: functional (black/white-box), boundary-value, equivalence partitions, decision-table testing
- Regression automated: existing regression suite fully automated (452 scripts). New features manually tested and automated in parallel.
- Exploratory sessions conducted for UX and edge cases.

Test Execution Model

- Combined manual + automation: manual for new/complex scenarios; automation for regression and repeatable flows.
- Continuous testing in CI for critical smoke/regression flows.

---

4. Test Environment Details (page 6)

Platforms & Browsers

- Desktop OS: Windows 10, macOS Ventura
- Mobile OS: Android 10–14, iOS 15–18
- Browsers: Chrome, Firefox, Edge, Safari, UC Browser

Infrastructure & Services

- Web Server: Apache Tomcat 10
- Database: MySQL 8.0
- Payment gateways: Existing + 5 new methods (note: enhanced monitoring required)
- Cloud/device matrix: BrowserStack App Live used for wide device coverage

Tools & Frameworks

- Test Management: TestLink, TestRail, Jira/GitHub issues
- Automation (UI): Selenium WebDriver, Cypress
- Mobile Automation: Appium 2.2.1 (Python 3.11)
- Performance: JMeter 5.6
- Static Analysis/Security: SonarQube
- Accessibility: Accessibility Scanner, VoiceOver/TalkBack
- CI: (project CI used to run automated suites — integrate regression in pre-deploy gating)

---

5. Test Execution Summary (page 7)

Execution Summary (overall)

- Total Planned Test Cases: 34
- Total Executed: 34 (100%)
- Passed: 24 → 70.6%
- Failed: 10 → 29.4%
- Blocked: 0

Automation & Coverage

- Automated tests executed in CI/automation runs: 45 automated tests in the new/combined suite
- Regression automated cases: 452 (existing regression pack)

- Automation coverage reported: 72% (45/63 as recorded in automation artifacts)
- Unit/Integration Code Coverage (from coverage reports): 87%

Critical Journeys

- Verified and passed: core critical journeys (login, scheduling creation, checkout, payment) for happy-paths in test environment. Note: some critical defects (session timeout, scheduling past-date validation) require fixes.

Sample Execution Table (excerpt)

Metric	Value
Total Planned	34
Executed	34
Passed	24
Failed	10
Automation Coverage (reported)	72% (45/63)
Unit Coverage	87%
Critical Journeys Passed	15/15 (happy-path verification)

Note on metrics consistency: original source artifacts had a few inconsistent percentage labels; the numbers above are normalized to match counts.

Screenshots & Evidence

- Automation logs, unit coverage screenshots and other attachments referenced in Appendices.

---

6. Defect Analysis & Categorization (page 9)

Defect Summary (top-level)

- Total distinct defects highlighted in QA: 9 (plus referenced enhancement)
- Severity distribution:
  - Critical: 1
  - High: 3
  - Medium: 3
  - Low: 1
  - Enhancement: 1

Detailed Defects

- ID 1 — Numerical name accepted
    - Severity: Medium | Status: Open | Note: Data integrity risk on user profiles.
- ID 2 — Can schedule pickup for past date
    - Severity: High | Status: Open | Note: Business/ops confusion; scheduling validation missing.
- ID 3 — Session timeout not implemented
    - Severity: Critical | Status: Open | Note: Security & usability risk; sessions persist beyond expected TTL.
- ID 4 — Dark mode missing on Awareness page
    - Severity: Enhancement | Status: Open | Note: UX improvement request.
- ID 5 — Accessibility alternatives missing
    - Severity: Medium | Status: Open | Note: Images/graphics lack alt text and some ARIA labels missing.
- ID 6 — Logout does not clear credentials
    - Severity: Medium | Status: Open | Note: Security/privacy (browser storage/session not cleared fully).
- ID 7 — Missing data in system statistics
    - Severity: Low | Status: Open | Note: Reporting accuracy; intermittent aggregation issue.
- ID 8 — Admin cannot use Edit in requests
    - Severity: High | Status: Open | Note: Admin functionality blocked; affects operations.
- ID 9 — Filter does not combine properly
    - Severity: High | Status: Open | Note: Filter combination logic incorrect — workflow disruption.

Top Root Causes Observed

- Missing validation rules (scheduling and input sanitation)
- Session lifecycle and token management gaps
- Partial client-server sync issues (wishlist sync)
- Incomplete accessibility attributes and UI state handling

Recommendations per Defect

- Critical/High: immediate triage and hotfix or blocked release gating until fixed or mitigated.
- Medium/Low: schedule for next sprint with ownership and test verification plan.
- Enhancement: backlog for UX sprint.

---

7. Risk Assessment (page 11)

Risk Summary & Matrix (high-level)

- RS_001 — Login: probability High, impact High → Critical
- RS_002 — Scheduling past date: probability Very High, impact Very High → Very High/Critical
- RS_003 — Session timeout missing: probability High, impact High → Critical
- RS_004 — Admin Edit blocked: probability Very High, impact Very High → Critical
- RS_005 — Accessibility gaps: probability Medium, impact Medium → Moderate
- RS_006 — Browser compatibility (Firefox/UC): probability Very High, impact Medium → Severe
- RS_007 — Logout/Credential persistence: probability Medium, impact Medium → Moderate

Risk Matrix (qualitative)

- Impact vs Probability mapping used to prioritize fixes and to decide release gating and monitoring strategy.

Release Risk Recommendation

- Do NOT proceed with a full open release until Critical issues (session timeout, scheduling past date, admin edit) are either fixed or a compensating control (feature flag, restricted rollout, enhanced monitoring/alerting & rapid rollback plan) is in place.
- If business requires release now, require an immediate hotfix plan + phased rollout (10% users), plus mandatory monitoring and rollback readiness.

Mitigation Actions

- Immediate hotfix for SHOP-4589 (wishlist sync) post-release or pre-release depending on release stance.
- Implement session TTL and server-side enforcement.
- Input validation on schedule creation (block dates < now).
- Add feature flags for new payment gateway activation and rollout by cohort.
- Add expanded telemetry/alerts (login failures, payment errors, server-side exceptions).

---

8. Recommendations & Improvements (page 13)

Immediate (Next 48–72 hours)

- Fix and deploy patches for critical defects: session timeout (ID3), scheduling validation (ID2), admin edit (ID8).
- Prepare hotfix for wishlist sync (SHOP-4589) and plan immediate verification after deployment.
- Enable enhanced monitoring (APM/tracing, payment gateway metrics, auth flows) for first 72 hours of rollout.

Short-term (Sprint-level)

- Resolve high & medium defects; include regression tests and automation for each fix.
- Improve logout flow to ensure credentials/session cleared client and server-side.
- Add server-side validation preventing scheduling in the past.

Medium-term (2–6 sprints)

- Accessibility: complete WCAG fixes (alt text, ARIA roles, screen-reader navigation). Target AA compliance.
- Expand automation coverage: target >90% regression coverage and add automated accessibility checks to CI.
- Browser compatibility fixes for Firefox and UC Browser.

Process Improvements

- Add pre-release checklist gating critical security and scheduling validations.
- Maintain a canary/feature-flagged release path with cohort-based monitoring.
- Expand A/B or canary monitoring to include synthetic tests for payment/auth every 5–10 minutes.

---

9. Test Metrics & KPIs (page 15)

Key Metrics Tracked

- Test case execution: 34/34 executed (100%)
- Pass rate (by case count): 24/34 → 70.6%
- Defect density: (9 defects recorded across suite) — use per-module breakdown for next report
- Automation coverage: 72% (as reported in automation artifacts; regression pack contains 452 scripts)
- Code coverage: 87% (unit/integration)
- Critical journeys status: 15/15 happy-path checks passed, but failing edge-case validations present
- Time to detect (TTD) critical defects: average 1.2 days during test cycle
- Time to fix (TTF) estimates (to be tracked with dev): Critical fixes target <72 hours

Suggested KPIs for Post-Release

- Production error rate (per 1k sessions) for first 72 hours — threshold for rollback set to X (agree with stakeholders)
- Payment failure rate delta from baseline
- Login/auth failure rate and average session duration
- Recovery time objective (RTO) for critical bug fixes ≤ 24–72 hours

---

10. Appendices (page 17)

A. Evidence & Attachments

- Automation logs & coverage screenshot references:
  - assets/unit-test-screenshot.png
  - automation logs: link_to_automation_logs.txt (internal)
  - coverage report: link_to_coverage_report.html (internal)
- Screenshots included during QA (as captured in original attachments):
  - https://github.com/user-attachments/assets/4b12fa12-c8a5-47da-8d67-7cf0784948ce
  - https://github.com/user-attachments/assets/ee92012e-2d77-4c47-a4dc-b4c9a0b32af2
  - https://github.com/user-attachments/assets/04115f6a-a603-4235-9f1b-1fb953162aa2
  - https://github.com/user-attachments/assets/4287e42c-1bab-4c17-9998-483792c67669
  - https://github.com/user-attachments/assets/91f7aa31-ebbd-4c66-b993-2d3582871172
  - https://github.com/user-attachments/assets/a5e050c7-d3e2-44f7-b327-e100590b7c08

B. Sample Test Cases (selected)

- TC_LG_001 — Login Functionality
  - Steps: Open app → Navigate to Login → Enter valid email/password → Submit
  - Test Data: Email user@cleancity.com / Password password123
  - Expected: Login successful and user redirected to dashboard
  - Actual: Login successful — Pass
- TC_WS_002 — Wishlist Synchronization
  - Steps: Add items to wishlist on device A → Remove item on device B → Refresh device A
  - Test Data: Multiple product items
  - Expected: Wishlist state consistent across devices
  - Actual: Intermittent loss observed — Fail (SHOP-4589)
- TC_AC_001 — Accessibility (Screen Reader)
  - Steps: Run VoiceOver/TalkBack across main user flows
  - Expected: All interactive content announced and navigable
  - Actual: Partial coverage; missing alt texts/labels — Partial

C. Issue Tracker & References

- Issues logged in GitHub: https://github.com/mah-c/wk-6-mah-c-1-glitch-grabbers-team-repo/issues (team repo for demo)
- Referenced ticket: SHOP-4589 — Wishlist synchronization hotfix

D. Glossary & Notes

- "Critical journey": business-critical path (e.g., login → schedule → checkout → payment)
- Percentages normalized where inconsistencies found in source artifacts
- Any URLs above may require internal access or permissions.

Prepared by: Meseret Akalu (CleanCity QA Team)
Reviewed by: Mercy Benu, Viron Ochieng

If you'd like, I can:

- Produce a printable PDF version (with proper pagination/headers) from this Markdown, or
- Export this report into a GitHub repo file and create initial GitHub issue templates for the high/critical defects, or
- Create a prioritized action plan (Jira/GitHub issue checklist) with owners and sprint estimates.

Which of those would you like me to do next?