QA TEST PLAN

Document Owner: QA / Customer Support
Last Updated: 12 January 2026
Target Release: Continuous / Sprint-based releases
Test Environment: Staging (Production-like)

This QA Test Plan documents the strategy, scope, and approach for testing a Trading Journal Application across web and mobile platforms. All scenarios are NDA-safe and representative of real production workflows. No confidential data or proprietary implementation details are included.

---

1. Purpose and Objectives

Purpose

The purpose of this test plan is to ensure that users can reliably log trades, review accurate analytics, sync data across devices, and export reports without data loss, calculation errors, or confusing user experience.

Objectives

- Validate data integrity for trade creation, editing, and deletion
- Ensure analytics calculations update correctly (PnL, win rate, risk metrics)
- Confirm data persistence and synchronization across sessions and devices
- Detect and prevent high-impact regressions during releases
- Provide clear defect reporting and release readiness status

---

2. Product Overview (High Level)

Application Type: Trading Journal (Web and Mobile)

Primary User Journeys:

- Create and edit trade records with notes and tags
- Review performance analytics and dashboards
- Search and filter historical trades
- Sync trade data across multiple devices
- Export trade data to CSV or PDF formats

---

3. Scope

In Scope (Core Features)

1. Trade Entry Management
   - Create, edit, and delete trades
   - Required field validation
   - Notes and tags management
   - Optional screenshot attachments
2. Trade History
   - List view and pagination or infinite scrolling
   - Filtering by date, strategy, or instrument
   - Search functionality
3. Analytics Dashboard
   - Total and period-based PnL
   - Win rate and average win/loss
   - Risk metrics (if supported)
   - Breakdown by tag, strategy, or instrument
4. Sync and Data Persistence
   - Session persistence
   - Cross-device synchronization
   - Graceful handling of poor or lost network connections
5. Exports
   - CSV export
   - PDF export (if supported)
   - Validation of data completeness and formatting
6. Notifications and Reminders (if supported)
   - Scheduled reminders
   - Time zone accuracy
   - User opt-in and opt-out controls

Out of Scope

- Live brokerage execution or order placement
- Real-time market data accuracy
- Payment processing or billing
- Backend infrastructure changes (handled by engineering)

---

4. Test Strategy

Test Levels

- Smoke Testing: Quick validation after deployment of critical flows
- Functional Testing: Feature-level validation against requirements
- Regression Testing: Ensure existing features are not broken
- Exploratory Testing: Identify edge cases and usability issues
- Support-driven Testing: Validate fixes based on recurring customer issues

---

Risk-Based Prioritization

Highest-risk areas include:

- Trade data not saving correctly
- Incorrect analytics calculations
- Sync failures or duplicate entries
- Incomplete or incorrect exports

Priority order for testing:

1. Trade creation, editing, and deletion
2. Analytics recalculation accuracy
3. Sync and data persistence
4. Filters and search
5. Export functionality
6. Notifications and reminders

---

5. Test Environments and Test Data

Test Environments

- Staging (production-like) – primary QA environment
- Production – verification only when permitted

Supported Platforms

Web:

- Chrome (latest)
- Firefox (latest)
- Edge (if required by user base)

Mobile:

- Android (latest and one prior version)
- iOS (latest and one prior version)

Test Data Setup

Use controlled and sanitized test accounts with the following datasets:

- Small dataset: 5–10 trades
- Medium dataset: 100–200 trades
- Large dataset: 1,000+ trades (for performance spot checks)

Data variations include:

- Different instruments (stocks, crypto, forex)
- Different tags and strategies
- Different currencies and time zones (if applicable)

---

6. Test Types and Coverage

Smoke Testing

Typical smoke test coverage includes:

- Login and dashboard access
- Create a trade with required fields

- Edit trade notes
- Delete a trade
- Verify analytics update
- Export basic CSV
- Refresh or relaunch app and confirm data persistence

Functional Testing

Functional testing focuses on:

- Validation rules for trade entry
- Trade history filtering and search logic
- Accuracy of analytics calculations (spot checks)
- Export file structure and completeness
- Notification trigger and settings behavior

Regression Testing

- Execute full regression checklist before release
- Focus on recently changed or high-risk modules
- Document pass/fail status and blockers

Exploratory Testing

Exploratory testing is time-boxed and includes:

- Boundary values (long notes, unusual symbols)
- Error handling and UX behavior
- Network interruption scenarios
- Multi-device sync sequences

Non-Functional Checks (Lightweight)

- Performance: Large trade list scrolling and dashboard load time
- Usability: Clear error messages and confirmations

- Reliability: No duplication or missing data after retries

---

7. Defect Management Process

Bug Report Contents

Each bug report includes:

- Clear title
- Environment details (platform, version, build)
- Steps to reproduce
- Expected result
- Actual result
- Severity and priority
- Screenshots or videos (if applicable)

Severity Definitions

- Critical: Data loss, crashes, unusable core functionality
- High: Major feature broken or frequent sync failures
- Medium: Incorrect behavior with workaround available
- Low: Cosmetic or minor UI issues

Triage Workflow

1. QA verifies and logs the defect
2. Severity and priority assigned
3. Engineering reviews and fixes
4. Fix deployed to staging
5. QA retests and verifies
6. Regression coverage updated if required

---

8. Entry and Exit Criteria

Entry Criteria

- Feature deployed to staging
- Change list or release notes available
- Test accounts and baseline data prepared

Exit Criteria

- No open Critical or High severity defects
- Smoke and regression tests completed
- Known issues documented with mitigation
- Release sign-off communicated to stakeholders

---

9. Release Testing and Reporting

Pre-release Checklist

- Confirm build or version
- Execute smoke tests
- Run targeted regression tests
- Validate high-risk areas

Reporting Summary

- Build/version tested
- Test scope completed
- Defect summary (opened/closed)
- Go or No-Go recommendation

---

10. Roles and Communication

QA / Customer Support:

- Execute test plan

- Report and verify defects
- Provide release readiness status
- Share customer issue insights

Engineering:

- Fix defects
- Provide logs and reproduction support
- Review release risks

Product / Stakeholders:

- Confirm acceptance criteria
- Approve release readiness

---

11. Known Risk Areas and Mitigations

Risk: Analytics not recalculating after trade updates
Impact: Incorrect user insights
Mitigation: Regression tests and calculation spot checks

Risk: Sync conflicts or duplicate trades
Impact: Data inconsistency
Mitigation: Multi-device sync testing

Risk: Export missing fields
Impact: User reporting issues
Mitigation: Validate columns and sample rows

Risk: Network interruptions
Impact: Partial saves or confusion
Mitigation: Test offline and retry scenarios

---

12. Continuous Improvement

Planned improvements include:

- Expanding regression coverage for analytics
- Adding lightweight API validation where possible
- Introducing automation for smoke tests
- Tracking defect trends and recurring support issues