Test Cases

Test Cases

Generated by adpa-enterprise-framework-automation v3.2.0
Category: quality-assurance
Generated: 2025-07-14T21:02:17.169Z
Description: Detailed test cases and test scenarios

ADPA - Advanced Document Processing & Automation Framework

Test Cases Documentation

1. Test Case Overview

Purpose and Scope

This document provides comprehensive test cases for the ADPA (Advanced Document Processing & Automation) Framework, an

enterprise-grade Node.js/TypeScript platform for AI-powered document generation, project management, and standards-compliant business analysis.

Scope includes CLI, REST API, admin web interface, AI provider integrations, document workflows, enterprise integrations (Adobe, Confluence, SharePoint), security, compliance, and performance.

Naming Conventions & Identification

- **Format:** `[Category]-[Module/Feature]-[Number]` (e.g., `FUNC-API-001`)
- **Categories:**
    - FUNC – Functional
    - UI – User Interface
    - INTEG – Integration
    - PERF – Performance
    - SEC – Security
    - NEG – Negative
    - REG – Regression

Test Case Categories & Priorities

- **Critical:** System/security core, data loss risk, compliance
- **High:** Major workflows, integrations, business rules
- **Medium:** UI/UX, non-critical flows, reporting
- **Low:** Cosmetic, usability, optional features

Traceability to Requirements

Each test case is mapped to one or more requirements from the product's documentation (README, guides, architecture, etc.), denoted as `REQ-XX` (see [Traceability Matrix](#) at end).

---

2. Functional Test Cases

FUNC-API-001: API Health Endpoint

- **Objective:** Ensure `/api/v1/health` returns accurate system health.
- **Preconditions:** API server running on `localhost:3001`.
- **Test Steps:**
    1. Send GET request to `http://localhost:3001/api/v1/health`.
- **Expected Results:**
    - HTTP 200 OK.
    - JSON body with `status`, `timestamp`, `version`, `environment`, `uptime`, `memory`, and `node`.
- **Test Data:** None.
- **Priority:** Critical.
- **Requirements Traceability:** REQ-API-HEALTH, README-API, BABOK-DEMO-1.2.

FUNC-CLI-001: Document Generation via CLI

- **Objective:** Verify CLI command generates a PMBOK Project Charter document.
- **Preconditions:** ADPA installed globally; output directory exists.
- **Test Steps:**
    1. Run `adpa generate --key project-charter --output ./docs`.
    2. Check for generated document in `./docs` folder.
- **Expected Results:**
    - Command executes without error.
    - Project Charter document present; content follows PMBOK 7th structure.
- **Test Data:** None.
- **Priority:** High.
- **Requirements Traceability:** REQ-CLI-GEN, README-CLI, PMBOK-REQ.

FUNC-TEMPLATE-001: Template Creation via API

- **Objective:** Validate creation of a new BABOK template.
- **Preconditions:** API server running; valid API key/JWT.
- **Test Steps:**
    1. POST to `/api/v1/templates` with valid template JSON (BABOK).
- **Expected Results:**
    - HTTP 201 Created.
    - JSON response includes template ID, name, and metadata.
- **Test Data:** Sample BABOK template JSON.
- **Priority:** High.
- **Requirements Traceability:** REQ-API-TEMPLATES, BABOK-DEMO-2.1.

FUNC-ADMIN-001: Access Admin Web Interface

- **Objective:** Ensure admin UI is accessible and displays dashboard.
- **Preconditions:** Admin interface running at `http://localhost:3001`.
- **Test Steps:**
    1. Open web browser at `http://localhost:3001`.
- **Expected Results:**
    - Login screen appears (if not authenticated).
    - After login, dashboard is visible with summary stats.
- **Test Data:** Admin credentials.
- **Priority:** High.
- **Requirements Traceability:** REQ-ADMIN-UI, README-ADMIN.

FUNC-SEC-001: API Key Authentication

- **Objective:** API endpoints require valid API key or JWT.
- **Preconditions:** API server running; valid and invalid API keys.
- **Test Steps:**
    1. Send request with valid API key in header `X-API-Key`.
    2. Send request with invalid API key.
    3. Send request with missing API key.

- **Expected Results:**
  - Valid: 200 OK, access granted.
  - Invalid: 401 Unauthorized, error message.
  - Missing: 401 Unauthorized, error message.
- **Test Data:** Known valid/invalid keys.
- **Priority:** Critical.
- **Requirements Traceability:** REQ-SEC-API, API-TESTING-2.

FUNC-ADOBE-001: Adobe PDF Generation

- **Objective:** System uses Adobe API to generate PDF from Markdown.
- **Preconditions:** Adobe credentials configured; sample Markdown file.
- **Test Steps:**
  1. Place Markdown file in input dir.
  2. Execute document generation (via CLI/API).
  3. Check output for PDF file.
- **Expected Results:**
  - PDF generated with correct content and styling.
  - Metadata correctly set.
- **Test Data:** Sample Markdown.
- **Priority:** High.
- **Requirements Traceability:** REQ-ADOBE-1, PHASE2-GUIDE.

---

3. User Interface Test Cases

UI-NAV-001: Admin UI Navigation

- **Objective:** Validate navigation/menu structure in admin UI.
- **Preconditions:** Admin server running; user logged in.
- **Test Steps:**
  1. Open admin UI.
  2. Click each sidebar/top menu item.

- **Expected Results:**
  - Correct page loads for each menu item.
  - No broken links.
- **Test Data:** N/A.
- **Priority:** Medium.
- **Requirements Traceability:** REQ-ADMIN-UI.

UI-FORM-001: Template Creation Form Validation

- **Objective:** Ensure form validation for new template creation.
- **Preconditions:** Admin UI, user logged in.
- **Test Steps:**
  1. Open "Create Template" form.
  2. Leave required fields empty; attempt to submit.
  3. Enter invalid data (special chars, too long).
- **Expected Results:**
  - Validation errors shown for empty/invalid fields.
  - Form cannot be submitted until valid.
- **Test Data:** Invalid field values.
- **Priority:** High.
- **Requirements Traceability:** REQ-UI-FORMVAL.

UI-RESP-001: Responsive Layout

- **Objective:** Admin UI renders correctly on desktop, tablet, mobile.
- **Preconditions:** Admin UI accessible.
- **Test Steps:**
  1. Open UI on desktop browser.
  2. Resize window to tablet and mobile widths.
  3. Open on actual mobile device.
- **Expected Results:**
  - Layout adapts; no horizontal scroll.
  - Menus/buttons accessible on all sizes.
- **Test Data:** N/A.
- **Priority:** Medium.

* **Requirements Traceability:** REQ-UI-RESP, README-FRONTEND.

UI-ACC-001: Accessibility Compliance

* **Objective:** UI meets WCAG 2.1 AA accessibility.
* **Preconditions:** Admin UI running.
* **Test Steps:**
    1. Run axe or Lighthouse accessibility audit.
    2. Navigate using only keyboard.
    3. Test with screen reader.
* **Expected Results:**
    * No critical accessibility issues.
    * All interactive elements reachable and labeled.
* **Test Data:** N/A.
* **Priority:** Medium.
* **Requirements Traceability:** REQ-UI-ACCESS.

---

4. Integration Test Cases

INTEG-CONFL-001: Confluence Publish

* **Objective:** Generated document publishes directly to Confluence.
* **Preconditions:** Confluence credentials/configured; Confluence integration initialized.
* **Test Steps:**
    1. Generate document (CLI/API).
    2. Run `adpa confluence publish --document ./docs/project-charter.md` .
    3. Check Confluence page.
* **Expected Results:**
    * Document appears in target Confluence space.
    * Metadata and formatting preserved.
* **Test Data:** Sample document.
* **Priority:** High.

- **Requirements Traceability:** REQ-CONFL-INTEG, README-CONFL.

INTEG-SHAREPOINT-001: SharePoint Upload

- **Objective:** Generated document uploads to SharePoint.
- **Preconditions:** SharePoint integration setup; Azure credentials.
- **Test Steps:**
    1. Run `adpa sharepoint upload --folder "Project Documents" --file ./docs/`.
    2. Verify document in SharePoint library.
- **Expected Results:**
    - Document present, correct folder.
    - Metadata and version applied.
- **Test Data:** Sample document.
- **Priority:** High.
- **Requirements Traceability:** REQ-SHAREPOINT, SHAREPOINT-GUIDE.

INTEG-AI-001: Multi-Provider AI Fallback

- **Objective:** AI provider failover works (e.g., OpenAI unavailable → Google AI).
- **Preconditions:** At least two AI providers configured; one intentionally disabled.
- **Test Steps:**
    1. Start document generation with primary provider offline.
    2. Observe logs/result.
- **Expected Results:**
    - System automatically switches to next available provider.
    - Document generation completes.
- **Test Data:** Provider API keys/settings.
- **Priority:** Critical.
- **Requirements Traceability:** REQ-AI-FAILOVER, ARCHITECTURE.

INTEG-DB-001: Database Operations

- **Objective:** All CRUD operations on templates/jobs persist to database (JSON/SQL).
- **Preconditions:** DB configured; API server running.
- **Test Steps:**
    1. Create, update, delete a template via API.
    2. List templates; verify persistence.
- **Expected Results:**
    - Data correctly written/updated/deleted.
    - API responses reflect DB state.
- **Test Data:** Template payloads.
- **Priority:** High.
- **Requirements Traceability:** REQ-DB, README-BACKEND.

5. Performance Test Cases

PERF-LOAD-001: API Load Testing

- **Objective:** API handles 100 concurrent generate requests.
- **Preconditions:** API server running; load test tool (e.g., JMeter).
- **Test Steps:**
    1. Send 100 concurrent POST requests to `/api/v1/generate`.
- **Expected Results:**

    - *95% requests complete within specified SLAs (e.g., <5s).*

    - No server crashes.
- **Test Data:** Standard generate payload.
- **Priority:** Critical.
- **Requirements Traceability:** REQ-PERF-API, README-PERF.

PERF-STRESS-001: Stress Test Document Generation

- **Objective:** System remains stable under extreme load (e.g., 500 simultaneous jobs).
- **Preconditions:** High-capacity environment.
- **Test Steps:**
    1. Initiate 500 document generation jobs.
    2. Monitor resource usage and error rates.
- **Expected Results:**
    - Graceful degradation (rate limit, queue, or error messages), no data loss.
- **Test Data:** Batch input.
- **Priority:** High.
- **Requirements Traceability:** REQ-PERF-STRESS.

PERF-RESP-001: Response Time

- **Objective:** API responds to `/api/v1/health` in <500ms under normal load.
- **Preconditions:** Normal load.
- **Test Steps:**
    1. Send repeated GET requests to `/api/v1/health`.
    2. Measure response time.
- **Expected Results:**
    - Median response time <500ms.
- **Test Data:** N/A.
- **Priority:** Medium.
- **Requirements Traceability:** REQ-PERF-RESP.

---

6. Security Test Cases

SEC-AUTH-001: User Authentication (JWT)

- **Objective:** Only authenticated users access protected endpoints.

- **Preconditions:** User accounts exist.
- **Test Steps:**
  1. Obtain JWT via login.
  2. Access protected API with JWT.
  3. Access API without JWT.
- **Expected Results:**
  - With JWT: Access granted.
  - Without JWT: 401 Unauthorized.
- **Test Data:** User credentials.
- **Priority:** Critical.
- **Requirements Traceability:** REQ-SEC-AUTH, README-SEC.

SEC-AUTHZ-001: Authorization - Role Checks

- **Objective:** Only users with admin role can manage templates.
- **Preconditions:** Users with admin and non-admin roles.
- **Test Steps:**
  1. Non-admin attempts to create/edit template.
  2. Admin creates/edits template.
- **Expected Results:**
  - Non-admin: 403 Forbidden.
  - Admin: 200 OK.
- **Test Data:** User roles.
- **Priority:** High.
- **Requirements Traceability:** REQ-SEC-AUTHZ.

SEC-INPUT-001: Input Validation & Sanitization

- **Objective:** System rejects malicious input (e.g., XSS, SQLi).
- **Preconditions:** API server running.
- **Test Steps:**
  1. Submit payload with `<script>` tags, SQL code, or oversized fields.
- **Expected Results:**
  - Input rejected, descriptive error.

- No code execution or DB corruption.
- **Test Data:** Malicious payloads.
- **Priority:** Critical.
- **Requirements Traceability:** REQ-SEC-INPUT.

SEC-VULN-001: Security Vulnerability Scan

- **Objective:** Detect known vulnerabilities (e.g., via npm audit, OWASP ZAP).
- **Preconditions:** Latest codebase.
- **Test Steps:**
    1. Run security scan tools.
- **Expected Results:**
    - No critical vulnerabilities.
    - All high/medium mitigated or documented.
- **Test Data:** N/A.
- **Priority:** High.
- **Requirements Traceability:** REQ-SEC-VULN.

---

7. Negative Test Cases

NEG-CLI-001: CLI Invalid Command

- **Objective:** CLI returns helpful error for unknown commands.
- **Preconditions:** CLI installed.
- **Test Steps:**
    1. Run `adpa generatee --key test`.
- **Expected Results:**
    - Error message: "Unknown command".
    - Usage/help displayed.
- **Test Data:** N/A.
- **Priority:** Medium.
- **Requirements Traceability:** REQ-NEG-CLI.

NEG-API-001: API Invalid Payload

- **Objective:** API rejects malformed or incomplete JSON on POST.
- **Preconditions:** API server running.
- **Test Steps:**
    1. POST to `/api/v1/templates` with missing required fields.
- **Expected Results:**
    - 400 Bad Request; error message describes missing fields.
- **Test Data:** Malformed JSON.
- **Priority:** High.
- **Requirements Traceability:** REQ-NEG-API.

NEG-SEC-001: Unauthorized File Upload

- **Objective:** System blocks upload of executable files via document upload endpoint.
- **Preconditions:** API server running.
- **Test Steps:**
    1. Attempt to upload `.exe` or `.js` file.
- **Expected Results:**
    - Upload rejected; error message shown.
- **Test Data:** Executable file.
- **Priority:** Critical.
- **Requirements Traceability:** REQ-NEG-FILEUPLOAD.

NEG-BOUND-001: Boundary Value Test - Field Length

- **Objective:** System enforces max character limits on fields (e.g., template name).
- **Preconditions:** API server running.
- **Test Steps:**
    1. Submit template with name of 256 characters (assuming max 255).
- **Expected Results:**
    - 400 Bad Request; error message.

- **Test Data:** Overlong name.
- **Priority:** Medium.
- **Requirements Traceability:** REQ-NEG-BOUND.

---

8. Regression Test Cases

REG-CORE-001: Core Document Generation

- **Objective:** Validate document generation after code changes.
- **Preconditions:** Regression suite; known good inputs.
- **Test Steps:**
    1. Generate all document types (BABOK, PMBOK, DMBOK).
    2. Compare output to baseline.
- **Expected Results:**
    - Output matches expected templates.
- **Test Data:** Baseline input/output.
- **Priority:** Critical.
- **Requirements Traceability:** REQ-REG-CORE.

REG-DEFECT-001: Retest Fixed Defect

- **Objective:** Confirm previously fixed defect does not reoccur.
- **Preconditions:** Defect test scenario.
- **Test Steps:**
    1. Execute steps from resolved defect ticket.
- **Expected Results:**
    - Defect not reproducible.
- **Test Data:** Relevant to defect.
- **Priority:** High.
- **Requirements Traceability:** REQ-REG-DEFECT.

REG-INTEG-001: Integration Points

- **Objective:** Integration with Adobe, SharePoint, Confluence continues to function.
- **Preconditions:** All integrations enabled.
- **Test Steps:**
    1. Run full document generation and publish flow.
- **Expected Results:**
    - All integrations succeed; no errors.
- **Test Data:** End-to-end scenario.
- **Priority:** Critical.
- **Requirements Traceability:** REQ-REG-INTEG.

REG-PERF-001: Performance Regression

- **Objective:** System performance is not degraded after updates.
- **Preconditions:** Baseline performance metrics.
- **Test Steps:**
    1. Repeat performance tests (PERF-LOAD-001, PERF-RESP-001).
- **Expected Results:**
    - Metrics within 10% of baseline.
- **Test Data:** Standard test suite.
- **Priority:** High.
- **Requirements Traceability:** REQ-REG-PERF.

REG-CONFIG-001: Configuration Change Impact

- **Objective:** Changes to `.env` or provider settings do not break workflows.
- **Preconditions:** Environment changes.
- **Test Steps:**
    1. Modify provider API keys, restart service.
    2. Run main workflows.
- **Expected Results:**
    - System operates as expected; no errors due to config changes.
- **Test Data:** Varied config files.
- **Priority:** Medium.

- **Requirements Traceability:** REQ-REG-CONFIG.

---

9. Test Data Requirements

- **Sample Documents:** Markdown, JSON templates for BABOK, PMBOK, DMBOK.
- **User Credentials:** Admin, business analyst, read-only accounts.
- **API Keys & Tokens:** For ADPA, Adobe, Confluence, SharePoint, AI providers (OpenAI, Google AI).
- **Malicious Payloads:** For security/negative testing (XSS, SQLi, oversized fields).
- **Files:** Valid PDFs, images, and invalid/executable files for upload scenarios.
- **Cleanup:**
  - Remove test documents from output, SharePoint, Confluence.
  - Reset database to pre-test state if required.
- **Privacy/Security:**
  - No production PII in test data.
  - Secure handling and deletion of credentials/test outputs.

---

10. Test Case Execution Guidelines

- **Execution Order:**
  1. Environment/health checks
  2. Security/authentication
  3. Core functional flows
  4. UI/UX
  5. Integrations
  6. Performance
  7. Negative/boundary
  8. Regression
- **Dependencies:**
  - API/DB must be running for API/UI tests
  - Credentials/configuration for integrations

- **Environment Prerequisites:**
    - Node.js 18+, TypeScript 5.7+, dependencies installed
    - Test/staging environment with all integrations enabled
- **Test Result Documentation:**
    - Log each test case: Pass/Fail, evidence/screenshots/logs, issues found
    - Record environment, test data, and steps taken
- **Defect Reporting:**
    - Log all failures in issue tracker (GitHub Issues/Jira)
    - Include steps, data, logs, and screenshots
    - Assign priority/severity and link to test case ID

Traceability Matrix (Excerpt)

Test Case ID	Requirement(s)
FUNC-API-001	REQ-API-HEALTH, README-API
FUNC-CLI-001	REQ-CLI-GEN, README-CLI
FUNC-TEMPLATE-001	REQ-API-TEMPLATES, BABOK-DEMO
FUNC-ADMIN-001	REQ-ADMIN-UI, README-ADMIN
INTEG-CONFL-001	REQ-CONFL-INTEG, README-CONFL
INTEG-SHAREPOINT-001	REQ-SHAREPOINT, SHAREPOINT-GUIDE
SEC-AUTH-001	REQ-SEC-AUTH, README-SEC
SEC-INPUT-001	REQ-SEC-INPUT
NEG-API-001	REQ-NEG-API

Test Case ID	Requirement(s)
REG-CORE-001	REQ-REG-CORE
...	...

End of Document

This test cases document is designed for direct use in both manual and automated test execution in enterprise environments.