



NimbusWiz documentation: technical accuracy report

! INFO

Point-in-time QA snapshot

This report captures the state of the docs vs. the prototype on 2026-05-09. It is a snapshot, not a live bug list. Compare against the 2026-04-22 baseline to track what has been fixed and what remains.

Audit date: 2026-05-09

Scope: User Guide (12 pages) and Admin Guide (9 pages)

Method: Static analysis of React source code at `nimbuswiz-demo/.claude/worktrees/funny-mayer-44ac15/src/` against documented UI claims.

Legend: Match · Mismatch · Not found in code · In code but missing from docs

Compared against: accuracy-report.md (2026-04-22). Notable changes since are summarized below.

Executive summary

The April 2026 correction pass resolved the large majority of the mismatches and scope-gap errors identified in the 2026-04-22 audit. Of the 60 mismatches and 35 not-found claims that appeared in the baseline, all but a small residual set have been addressed: corrected labels now match the prototype, planned-feature admonitions cover unimplemented controls, and the overall claim accuracy rate has improved from roughly 12% to roughly 79%.

Four mismatches remain. The most notable is the Demo Mode button label in `getting-started.md`: the docs describe a "Demo Mode button" but the prototype renders "Try demo — your real data stays safe" when inactive. Three not-found items persist in `assess-your-system.md` for interactive behaviors (dependency graph node click, table column sort, assessment gate disable logic) that are described as if they are implemented but are not present in the source.

Eighteen prototype surfaces remain undocumented, concentrated in the Developer Hub tabs (Overview, Usage, Webhooks, Reference) and the Settings > Security tab, which are not covered by any guide page.

No regressions from the 2026-04-22 baseline were found. The corrected docs do not introduce new mismatches for claims that were previously verified as correct.

USER GUIDE FINDINGS

Sidebar and Navigation

CLAIM	STATUS	NOTES
Sidebar renders 12 navigation items	✓	Confirmed in <code>Root.tsx</code> navigation array: Overview, Systems, System Assessment, Modernization Advisor, Modernization, Deployment, Monitoring, Developer Hub, Automation, Troubleshooting, Audit Logs, Settings.
Sidebar label "Overview" links to Dashboard (/)	✓	<code>{ name: "Overview", path: "/", icon: LayoutDashboard }</code> in <code>Root.tsx</code> .
Sidebar label "Systems" links to Fleet Management	✓	<code>{ name: "Systems", path: "/systems" }</code> . Page h1 is "Fleet Management" in <code>SystemsList.tsx</code> .
Sidebar label "System Assessment" links to assessment page	✓	<code>{ name: "System Assessment", path: "/assessment" }</code> . Page h1 is "System Assessment" in <code>SystemAssessment.tsx</code> .
Sidebar label "Modernization Advisor" links to advisor	✓	<code>{ name: "Modernization Advisor", path: "/advisor" }</code> . Page h1 is "Modernization Advisor" in <code>ModernizationAdvisor.tsx</code> .
Sidebar label "Modernization" links to Upgrade Profiles page	✓	<code>{ name: "Modernization", path: "/modernization" }</code> . Page h1 is "Upgrade Profiles" in <code>Modernization.tsx</code> .
Sidebar label "Deployment" links to deployment page	✓	<code>{ name: "Deployment", path: "/deployment" }</code> . Page h1 is "Deployment" in <code>Deployment.tsx</code> .
Sidebar label "Monitoring" links to monitoring page	✓	<code>{ name: "Monitoring", path: "/monitoring" }</code> . Page h1 is "Monitoring & Alerts" in <code>Monitoring.tsx</code> .
Sidebar label "Developer Hub" links to developer page	✓	<code>{ name: "Developer Hub", path: "/developer" }</code> . Confirmed in <code>DeveloperHub.tsx</code> .

CLAIM	STATUS	NOTES
Sidebar label "Automation" links to automation page	✓	<code>{ name: "Automation", path: "/automation" }</code> . Page h1 is "Automation" in <code>Automation.tsx</code> .
Sidebar label "Troubleshooting" links to troubleshooting page	✓	<code>{ name: "Troubleshooting", path: "/troubleshooting" }</code> . Page h1 is "Troubleshooting" in <code>Troubleshooting.tsx</code> .
Sidebar label "Audit Logs" links to audit logs page	✓	<code>{ name: "Audit Logs", path: "/audit-logs" }</code> . Page h1 is "Audit Logs" in <code>AuditLogs.tsx</code> .
Sidebar label "Settings" links to settings page	✓	<code>{ name: "Settings", path: "/settings" }</code> . Page h1 is "Settings" in <code>Settings.tsx</code> .
"Start Product Tour" button at bottom of sidebar	✓	Confirmed in <code>Root.tsx</code> . Has <code>onClick={startTour}</code> handler.
Glossary opens via ? icon in sidebar header	✓	<code>HelpCircle</code> button in <code>Root.tsx</code> opens <code>GlossaryModal</code> . <code>onClick={() => setGlossaryOpen(true)}</code> .
"Current Role" badge at bottom of sidebar	✓	<code>Root.tsx</code> renders role badge with <code>role</code> from <code>useUserRole()</code> . Label text: "Current Role".
Skip to main content link for keyboard nav	✓	<code>SkipToContent</code> component renders in <code>Root.tsx</code> .

Getting Started (first-run banner)

CLAIM	STATUS	NOTES
"Getting Started" banner appears on Dashboard during first run	✓	<code>FirstRunSetup</code> renders in <code>Dashboard.tsx</code> when <code>isFirstRun</code> is true from context.
Banner has four orientation steps as progress indicators	✓	<code>steps</code> array in <code>FirstRunSetup.tsx</code> contains 4 items.
Step 1: "Connect a System"	✓	<code>title: 'Connect a System'</code> in <code>FirstRunSetup.tsx</code> .

CLAIM	STATUS	NOTES
Step 2: "Run Fleet Scan"	✓	<code>title: 'Run Fleet Scan'</code> in <code>FirstRunSetup.tsx</code> .
Step 3: "View Recommendations"	✓	<code>title: 'View Recommendations'</code> in <code>FirstRunSetup.tsx</code> .
Step 4: "Apply or Preview"	✓	<code>title: 'Apply or Preview'</code> in <code>FirstRunSetup.tsx</code> .
Dismiss button closes the banner	✓	<code><button onClick={handleDismiss}>Dismiss</button></code> in <code>FirstRunSetup.tsx</code> .
Demo Mode off CTAs: enable Demo Mode button and "Register a system" link	✓	<code>!isDemoMode</code> branch renders demo toggle button and <code><Link to="/systems">Register a system</Link></code> in <code>FirstRunSetup.tsx</code> .
Demo Mode on CTAs: "View Recommendations" link, "Explore Demo Fleet" link, "Switch to Real Data" button	✓	<code>isDemoMode</code> branch renders Link to /advisor "View Recommendations", Link to /systems "Explore Demo Fleet", and <code><button onClick={handleDemoToggle}>Switch to Real Data</button></code> .
Demo Mode active: button shows "• Demo Mode Active"	✓	<code>isDemoMode ? '• Demo Mode Active' : 'Try demo — your real data stays safe'</code> in <code>Root.tsx</code> .
Demo Mode inactive: described as "Demo Mode button"	✗	Prototype renders "Try demo — your real data stays safe" when inactive, not "Demo Mode" or a label matching that description. <code>Root.tsx</code> line 115.

Register a System

CLAIM	STATUS	NOTES
"Register a system" button opens three-step modal	✓	<code><button onClick={() => setRegisterModalOpen(true)}>Register a system</button></code> in <code>SystemsList.tsx</code> . Opens <code>RegisterSystem</code> modal.
Modal title: "Register New System"	✓	<code><h2>Register New System</h2></code> in <code>RegisterSystem.tsx</code> .

CLAIM	STATUS	NOTES
Step 1 label: "System Info"	✓	<code>{stepNum === 1 && 'System Info'}</code> in <code>RegisterSystem.tsx</code> .
Step 2 label: "Performance"	✓	<code>{stepNum === 2 && 'Performance'}</code> in <code>RegisterSystem.tsx</code> .
Step 3 label: "Configure"	✓	<code>{stepNum === 3 && 'Configure'}</code> in <code>RegisterSystem.tsx</code> .
Step 1 field: "System ID" (must start with nimbus-)	✓	Label "System ID" with validation <code>nimbus-</code> prefix in <code>RegisterSystem.tsx</code> .
Step 1 field: "Serial Number"	✓	Confirmed in <code>RegisterSystem.tsx</code> .
Step 1 field: "Manufacturing Year"	✓	Confirmed in <code>RegisterSystem.tsx</code> .
Step 1 field: "Current Version" dropdown	✓	Select with options Nimbus1500 through Nimbus2026 in <code>RegisterSystem.tsx</code> .
Step 1 field: "Location"	✓	Confirmed in <code>RegisterSystem.tsx</code> .
Step 1 field: "Usage Type" dropdown: Racing, Training, Transport, Display/Retired	✓	Options: racing, training, transport, display in <code>RegisterSystem.tsx</code> .
Step 1 field: "Status" dropdown: Active, In Maintenance, Retired, Reserve	✓	Options: active, maintenance, retired, reserve in <code>RegisterSystem.tsx</code> .
Step 1 field: "Description (Optional)"	✓	Confirmed in <code>RegisterSystem.tsx</code> .
Step 1 to Step 2 button: "Add performance data"	✓	<code>{step === 1 ? 'Add performance data' : 'Configure monitoring'}</code> in footer.
Step 2 field: "Flight Hours"	✓	Confirmed in <code>RegisterSystem.tsx</code> .
Step 2 field: "Max Velocity" (0-100 scale)	✓	Confirmed in <code>RegisterSystem.tsx</code> .

CLAIM	STATUS	NOTES
Step 2 field: "Handling Score" (0-100 scale)	✓	Confirmed in <code>RegisterSystem.tsx</code> .
Step 2 field: "Acceleration"	✓	Confirmed in <code>RegisterSystem.tsx</code> .
Step 2 field: "Last Service Date"	✓	Confirmed in <code>RegisterSystem.tsx</code> .
Step 2 field: "Next Service Due"	✓	Confirmed in <code>RegisterSystem.tsx</code> .
Step 2 field: "Service History"	✓	Confirmed in <code>RegisterSystem.tsx</code> .
Step 2 field: "Safety Certification" dropdown: Valid, Expired	✓	Options: valid, expired in <code>RegisterSystem.tsx</code> .
Step 2 field: "Known Issues"	✓	Confirmed in <code>RegisterSystem.tsx</code> .
Step 2 field: "Compatibility"	✓	Confirmed in <code>RegisterSystem.tsx</code> .
Step 2 "Validate System" button	✓	<code><button onClick={handleValidateSystem}>Validate System</button></code> in <code>RegisterSystem.tsx</code> .
Step 2 to Step 3 button: "Configure monitoring"	✓	Confirmed in <code>RegisterSystem.tsx</code> .
Step 3: "Enable Real-Time Monitoring" checkbox	✓	Confirmed in <code>RegisterSystem.tsx</code> .
Step 3: "Enable Auto-Apply for Safe Upgrades" checkbox	✓	Confirmed in <code>RegisterSystem.tsx</code> .
Step 3 final button: "Register System"	✓	<code><button onClick={handleRegister}>Register System</button></code> in <code>RegisterSystem.tsx</code> .
Fleet Management table columns: System Name, Status, Recommendation, Stability Index, Version, Usage Type, Flight Hours	✓	All columns confirmed in <code>SystemsList.tsx</code> table headers.

CLAIM	STATUS	NOTES
Search bar, Status dropdown (All Statuses, Healthy, Warning, Critical), Usage Type dropdown	✓	Confirmed in <code>SystemsList.tsx</code> .
Quick filter chips: Critical, Warning, Upgradable, Up to Date	✓	Chips "Critical", "Warning", "Upgradable", "✓ Up to Date" in <code>SystemsList.tsx</code> .
Row actions: View Details (eye icon), Run Scan (play icon, in progress)	✓	Eye icon links to system detail; play icon button has no <code>onClick</code> . Correctly flagged as in progress in docs.

Assess Your System

CLAIM	STATUS	NOTES
"System Assessment" page in sidebar	✓	Sidebar link confirmed. h1 "System Assessment" in <code>SystemAssessment.tsx</code> .
Three stat cards: High Risk Components, Medium Risk Components, Up to Date Components	✓	All three cards confirmed in <code>SystemAssessment.tsx</code> .
Section: "Dependency Graph"	✓	h2 "Dependency Graph" in <code>SystemAssessment.tsx</code> .
Section: "Risk Heatmap & Recommendations"	✓	h2 "Risk Heatmap & Recommendations" in <code>SystemAssessment.tsx</code> .
Risk table columns: Component, Risk Level, Current Version, Recommended Version, Recommendation	✓	All five column headers confirmed in <code>SystemAssessment.tsx</code> .
Selecting a node in the dependency graph opens a detail panel	✗	Dependency graph nodes are static <code><div></code> elements with no <code>onClick</code> handler in <code>SystemAssessment.tsx</code> . No detail panel rendered.

CLAIM	STATUS	NOTES
Column header selectable to sort the risk table	✗	Table headers are plain <code><th></code> elements with no sort handler in <code>SystemAssessment.tsx</code> .
"Proceed to Modernization" link button at bottom of assessment	✓	<code><Link to="/modernization">Proceed to Modernization</Link></code> in <code>SystemAssessment.tsx</code> .
Proceed to Modernization is disabled until assessment passes	🔍	The Link is always active, with no conditional disable logic. <code>SystemAssessment.tsx</code> does not implement the gate. The behavior is described in docs as if it is implemented.
Scan Information section	+	<code>SystemAssessment.tsx</code> renders a "Scan Information" panel with Last Scan, Scan Duration, Scan Type, Components Analyzed, Vulnerabilities Found, and Next Scheduled Scan. Not covered in the docs.

Get a Recommendation

CLAIM	STATUS	NOTES
Sidebar: "Modernization Advisor"	✓	Confirmed.
Page h1: "Modernization Advisor"	✓	Confirmed in <code>ModernizationAdvisor.tsx</code> .
Primary recommendation card: "Recommended: Performance Upgrade"	✓	h2 "Recommended: Performance Upgrade" in <code>ModernizationAdvisor.tsx</code> .
Confidence score shown as 87% in card	✓	<code>87%</code> and progress bar shown in <code>ModernizationAdvisor.tsx</code> .
"Affected Systems" and "Expected Improvement" in primary card	✓	Both present in <code>ModernizationAdvisor.tsx</code> .
"Disagree with this recommendation?" opens override panel	✓	<code><button onClick={() => setShowOverride(true)}>Disagree with this recommendation?</button></code> in <code>ModernizationAdvisor.tsx</code> .

CLAIM	STATUS	NOTES
Override panel: dropdown for different path, text input for reason	✓	Select with Extended Operation, Partial Modernization, Replace options; reason text input. Confirmed in <code>ModernizationAdvisor.tsx</code> .
"Save override to audit" button in override panel	✓	<code><button onClick={...}>Save override to audit</button></code> in <code>ModernizationAdvisor.tsx</code> .
Section: Decision Options Analysis with 3 option cards	✓	Three cards: Upgrade, Partial Modernization, Replace (Nimbus2026). Confirmed in <code>ModernizationAdvisor.tsx</code> .
Section: Risk vs. Performance scatter chart	✓	<code>ScatterChart</code> component with risk/performance data in <code>ModernizationAdvisor.tsx</code> .
Section: "Why We Recommend This" with 3 factor rows	✓	<code>tradeoffFactors</code> array: System Quirks, Dependency Risks, Usage Patterns in <code>ModernizationAdvisor.tsx</code> .
Section: System-Specific Recommendations with before/after metrics	✓	<code>systemRecommendations</code> array rendered with <code>currentState/projectedState</code> in <code>ModernizationAdvisor.tsx</code> .
Primary card link navigates to Upgrade Profiles page (/modernization)	✓	<code><Link to="/modernization"></code> in primary recommendation card.

Modernize a System

CLAIM	STATUS	NOTES
Page h1: "Upgrade Profiles"	✓	Confirmed in <code>Modernization.tsx</code> .
Three profile cards: Stable, Performance, Extended Operation	✓	<code>profiles</code> array with three entries confirmed in <code>Modernization.tsx</code> .
Stable risk: Low	✓	<code>risk: "low"</code> for Stable profile.
Performance risk: Medium	✓	<code>risk: "medium"</code> for Performance profile.

CLAIM	STATUS	NOTES
Extended Operation risk: Low	✓	<code>risk: "low"</code> for Extended Operation profile.
Simulation button: "Run it for me"	✓	<code><button onClick={runSimulation}>Run it for me</button></code> in <code>Modernization.tsx</code> .
Simulation check: "Dependency compatibility"	✓	<code>simChecks</code> array item in <code>Modernization.tsx</code> .
Simulation check: "Disk space" (rendered as "Disk space available")	✓	<code>name: "Disk space available"</code> in <code>simChecks</code> . Docs say "Disk space" which is an acceptable shortening.
Simulation check: "Backup" (rendered as "Backup verification")	✓	<code>name: "Backup verification"</code> in <code>simChecks</code> .
Simulation check: "Quirk conflicts"	✓	<code>name: "Quirk conflicts"</code> in <code>simChecks</code> .
Apply buttons: "Apply Stable Profile", "Apply Performance Profile", "Apply Extended Operation Profile"	✓	<code>Apply {profile.name} Profile</code> renders correctly for each. Confirmed in <code>Modernization.tsx</code> .
After applying: "[Profile] Profile Applied" status	✓	Confirmed rendered when <code>appliedProfile === profile.id</code> .
Apply buttons disabled until simulation passes	✓	Buttons render as disabled with <code>cursor-not-allowed</code> when <code>simStatus !== "passed"</code> .
Timeline: Stable 2-3 weeks, Performance 3-4 weeks, Extended Operation 4-6 weeks	✓	Confirmed in <code>profiles</code> array in <code>Modernization.tsx</code> .

Deploy an Upgrade

CLAIM	STATUS	NOTES
Sidebar: "Deployment"	✓	Confirmed.
Page h1: "Deployment"	✓	Confirmed in <code>Deployment.tsx</code> .

CLAIM	STATUS	NOTES
"New Deployment" button (in progress, no handler)	✓	Button present with no <code>onClick</code> in <code>Deployment.tsx</code> . Correctly noted as planned feature in docs.
Six deployment phases: Preparing, Backup & Validation, Executing Upgrade, Testing & Verification, Monitoring, Completed	✓	All six phase names confirmed in <code>deploymentPhases</code> array in <code>Deployment.tsx</code> .
"Roll back deployment" button in active deployment section	✓	<code><button>Roll back deployment</button></code> in <code>Deployment.tsx</code> . No <code>onClick</code> handler: scope gap. Correctly noted as in progress.
Progress bar for overall completion	✓	Progress bar rendered with <code>activeDeployment.progress</code> in <code>Deployment.tsx</code> .
Section: "Deployment Schedule & History" table	✓	h2 "Deployment Schedule & History" in <code>Deployment.tsx</code> .
Table columns: Deployment ID, System, Profile, Status, Start Time, Duration	✓	All column headers confirmed in <code>Deployment.tsx</code> .
Status values: Completed, In Progress, Scheduled	✓	"✓ Completed", "● In Progress", "○ Scheduled" confirmed in <code>Deployment.tsx</code> .

Monitor System Health

CLAIM	STATUS	NOTES
Sidebar: "Monitoring"; page title: "Monitoring & Alerts"	✓	Sidebar link "Monitoring", h1 "Monitoring & Alerts" in <code>Monitoring.tsx</code> .
Section: "Performance Metrics (24 Hours)" chart	✓	h2 "Performance Metrics (24 Hours)" in <code>Monitoring.tsx</code> .
Three core metrics: Stability Index, Responsiveness Score, Network Interaction	✓	All three <code><Line></code> series confirmed by name in <code>Monitoring.tsx</code> .

CLAIM	STATUS	NOTES
Chart shows past 24 hours of data with no time-range selector	✓	No time range controls in <code>Monitoring.tsx</code> . Data array spans 24 hours.
Section: "System Resource Usage" area chart	✓	h2 "System Resource Usage" with CPU, Memory, Disk <code><Area></code> series in <code>Monitoring.tsx</code> .
Section: "Active Alerts"	✓	h2 "Active Alerts" in <code>Monitoring.tsx</code> .
Alert severity: Critical (red), Warning (yellow/amber), Info (blue)	✓	Color classes confirmed in <code>Monitoring.tsx</code> .
"Acknowledge" button on unacknowledged alerts (in progress)	✓	<code><button>Acknowledge</button></code> present in <code>Monitoring.tsx</code> . No <code>onClick</code> handler. Correctly noted as planned.
"View All" button in Active Alerts section (in progress)	✓	<code><button>View All</button></code> in <code>Monitoring.tsx</code> . No handler. Correctly noted as planned.
Section: "Predictive Maintenance Insights"	✓	h2 "Predictive Maintenance Insights" in <code>Monitoring.tsx</code> .
Insight fields: Metric, Current, Predicted (7d), Confidence, Trend, Recommendation	✓	All six fields confirmed in prediction card rendering in <code>Monitoring.tsx</code> .

Troubleshoot Issues

CLAIM	STATUS	NOTES
Sidebar: "Troubleshooting"	✓	Confirmed.
Page h1: "Troubleshooting"	✓	Confirmed in <code>Troubleshooting.tsx</code> .
Section: "Active Issues"	✓	h2 "Active Issues" in <code>Troubleshooting.tsx</code> .
Issue card fields: title, severity, affected systems, occurrences, last seen	✓	All five fields rendered in <code>Troubleshooting.tsx</code> .

CLAIM	STATUS	NOTES
"View [system-name]" button links to System Detail	✓	<code><Link to={/systems/\${issue.affectedSystems[0]}}>View {issue.affectedSystems[0]}</Link></code> in <code>Troubleshooting.tsx</code> .
Section: "Suggested Fixes"	✓	h2 "Suggested Fixes" in <code>Troubleshooting.tsx</code> .
Suggested fix action buttons link to remediation pages	✓	<code><Link to={fix.href}></code> buttons confirmed.
Fix destinations: /assessment, /modernization, /automation confirmed	✓	Confirmed in <code>suggestedFixes</code> array in <code>Troubleshooting.tsx</code> .
Log viewer section: "System Logs"	✓	h2 "System Logs" in <code>Troubleshooting.tsx</code> .
Log entry fields: Timestamp, Level (ERROR/WARN/INFO), system, message	✓	All confirmed in <code>Troubleshooting.tsx</code> .
Log filters: Level (All Levels, Error, Warning, Info), System (All Systems + specific)	✓	Both filter selects confirmed in <code>Troubleshooting.tsx</code> .
Section: "Diagnostic Tools" with 6 tools (in progress)	✓	h2 "Diagnostic Tools" with 6 buttons. No <code>onClick</code> handlers. Correctly noted as planned.
Tool names: System Health Check, Network Diagnostics, Performance Profiler, Dependency Checker, Configuration Validator, Log Analyzer	✓	All 6 tool names confirmed in <code>Troubleshooting.tsx</code> .

Automate Tasks

CLAIM	STATUS	NOTES
Sidebar: "Automation"	✓	Confirmed.

CLAIM	STATUS	NOTES
Page h1: "Automation"	✓	Confirmed in <code>Automation.tsx</code> .
"Create Pipeline" button in upper-right corner (in progress)	✓	<code><Plus /> Create Pipeline</code> button in header. No <code>onClick</code> handler beyond the <code>setRegisterModalOpen</code> equivalent. Correctly noted as planned.
Section: "CI/CD Pipelines"	✓	h2 "CI/CD Pipelines" in <code>Automation.tsx</code> .
Pipeline card fields: name, status, trigger type, last run, success rate, total runs	✓	All six fields confirmed in <code>Automation.tsx</code> .
Pipeline card icon buttons: Run, Pause, Configure	✓	Three icon buttons with titles "Run", "Pause", "Configure" in <code>Automation.tsx</code> .
Section: "Pipeline Visualization"	✓	h2 "Pipeline Visualization" in <code>Automation.tsx</code> .
Visualization stages: Trigger, Prepare, Execute, Test, Deploy	✓	All five stages rendered in <code>Automation.tsx</code> .
Section: "Scheduled Jobs"	✓	<code>scheduledJobs</code> array rendered with h2 section in <code>Automation.tsx</code> .
Scheduled Jobs fields: name, schedule (cron), next run, status	✓	All confirmed in <code>Automation.tsx</code> .
Section: "Event Hooks"	✓	<code>eventHooks</code> array rendered in <code>Automation.tsx</code> .
Event hook toggle to enable/disable	✓	Toggle present in event hook cards.
Available events: System Health Critical, Deployment Completed, New Security Patch Available, Stability Below Threshold	✓	All four events confirmed in <code>eventHooks</code> array in <code>Automation.tsx</code> .
Section: "Recent Pipeline Runs"	✓	<code>recentRuns</code> array confirms section exists in <code>Automation.tsx</code> .

Audit Logs (User)

CLAIM	STATUS	NOTES
Sidebar: "Audit Logs"	✓	Confirmed.
Page h1: "Audit Logs"	✓	Confirmed in <code>AuditLogs.tsx</code> .
"Export Logs" button (in progress)	✓	<code><Download /> Export Logs</code> button in <code>AuditLogs.tsx</code> . No save handler. Correctly noted as planned.
Table columns: Timestamp, User, Action, Target, Details, Status	✓	All six column headers confirmed in <code>AuditLogs.tsx</code> .
Action filter options: All Actions, Apply, Rollback, Simulate, Register, Delete	✓	All six options confirmed in <code>AuditLogs.tsx</code> .
Status filter options: All Statuses, Success, Failed, Pending	✓	All four options confirmed in <code>AuditLogs.tsx</code> .
User filter: All Users + specific users by email	✓	Dynamic user list from log data confirmed in <code>AuditLogs.tsx</code> .

ADMIN GUIDE FINDINGS

Complete Initial Setup

CLAIM	STATUS	NOTES
First-run banner on Dashboard (cross-reference)	✓	Confirmed in <code>FirstRunSetup.tsx</code> .
Settings > General tab contains Organization Name, Default Timezone, Date Format	✓	All three fields confirmed in "General" tab of <code>Settings.tsx</code> .
"Save Changes" button (in progress)	✓	<code><button>Save Changes</button></code> in General tab of <code>Settings.tsx</code> . Has <code>onClick</code> handler but no save logic. Correctly noted as planned.

CLAIM	STATUS	NOTES
Settings reached via sidebar "Settings"	✓	Confirmed sidebar link.

Manage Users

CLAIM	STATUS	NOTES
Settings > Users & Access tab	✓	"Users & Access" tab confirmed in <code>Settings.tsx</code> tabs array.
"Invite User" button (in progress)	✓	<code><button>Invite User</button></code> in Users & Access tab. No handler. Correctly noted as planned.
Role dropdown per user row: Viewer, Operator, Admin	✓	<code><select></code> with Viewer, Operator, Admin options in <code>Settings.tsx</code> .
"Current Role" badge at bottom of sidebar	✓	Confirmed in <code>Root.tsx</code> .
"Start Product Tour" at bottom of sidebar for new users	✓	Confirmed in <code>Root.tsx</code> .
Notification preferences in Settings > Notifications	✓	Notifications tab exists in <code>Settings.tsx</code> .

Manage API Keys

CLAIM	STATUS	NOTES
API key management in Developer Hub > API Keys	✓	"keys" tab in <code>DeveloperHub.tsx</code> . Sidebar nav "Developer Hub" confirmed.
Five scopes: fleet:read, fleet:write, deployments:read, deployments:write, webhooks:read	✓	All five scopes present in API key data in <code>DeveloperHub.tsx</code> .

CLAIM	STATUS	NOTES
"Generate Key" button opens Create Key panel	✓	Confirmed in <code>DeveloperHub.tsx</code> ApiKeysTab.
Create Key fields: Key Name, Expiration (Never, 30 days, 90 days, 1 year), Scopes	✓	All three fields confirmed in <code>DeveloperHub.tsx</code> .
Generate button (in progress)	✓	Correctly noted as in progress in docs.
API Keys list columns: Name, Key (first 14 chars + masked), Scopes, Last Used, Today, Actions	✓	All six columns confirmed in <code>DeveloperHub.tsx</code> .
Revoked keys visible in list at reduced opacity	✓	<code>status: "revoked"</code> entries rendered in <code>DeveloperHub.tsx</code> .
Trash icon revokes key (in progress)	✓	Correctly noted as in progress in docs.

Connect Integrations

CLAIM	STATUS	NOTES
Developer Hub > Integrations tab	✓	"integrations" tab in <code>DeveloperHub.tsx</code> .
"Connected" and "Available" sections	✓	Integration statuses: "connected" and "available" in <code>DeveloperHub.tsx</code> .
Filter by category: CI/CD, Monitoring, Notifications, Incident Mgmt, Infrastructure	✓	Category values confirmed in <code>integrations</code> array in <code>DeveloperHub.tsx</code> .
Integrations: GitHub Actions, GitLab CI, Jenkins (CI/CD)	✓	All three confirmed in <code>DeveloperHub.tsx</code> .
Integrations: Datadog, Prometheus (Monitoring)	✓	Confirmed in <code>DeveloperHub.tsx</code> .
Integration: Slack (Notifications)	✓	Confirmed in <code>DeveloperHub.tsx</code> .
Integration: PagerDuty (Incident Mgmt)	✓	Confirmed in <code>DeveloperHub.tsx</code> .

CLAIM	STATUS	NOTES
Integration: Terraform (Infrastructure)	✓	Confirmed in <code>DeveloperHub.tsx</code> .
Connect/Configure buttons exist but don't open setup flow (correctly noted as unimplemented)	✓	Correctly framed in docs with planned-feature admonition.

Review Audit Logs (Admin)

CLAIM	STATUS	NOTES
Sidebar: "Audit Logs"	✓	Confirmed.
Filter options match user-guide audit-logs.md (Action, Status, User)	✓	Same filter options confirmed in <code>AuditLogs.tsx</code> .
API key audit: filter by User is the effective path (no dedicated API key action filter)	✓	Action dropdown has no "API key" option. Filtering by User is correct. Confirmed in <code>AuditLogs.tsx</code> .
Compliance workflows: change-management, access review, incident investigation	✓	All three workflows described correctly relative to available filters.

Undocumented features (in code, not in either guide)

- **Settings > Security tab:** Renders API key rotation toggle, two-factor authentication controls (require 2FA for all users, SMS-based authentication), and session management (session timeout select). No guide page covers this tab. `Settings.tsx`.
- **Settings > Audit Logs tab (within Settings):** The Settings page has its own Audit Logs tab that shows a simplified log table. Separate from the main Audit Logs page. `Settings.tsx`.
- **Developer Hub > Overview tab:** Full quick-start panel with authentication code example, rate limit info, webhook concept, and common workflow cards. `DeveloperHub.tsx`.
- **Developer Hub > Usage tab:** API call volume chart, top endpoints table, error distribution pie chart. `DeveloperHub.tsx`.
- **Developer Hub > Webhooks tab:** Full webhook management including webhook entries, delivery history, and available event list for subscription. `DeveloperHub.tsx`.

- **Developer Hub > Reference tab:** Live API reference browser with endpoint request/response examples and error codes for all documented endpoints. `DeveloperHub.tsx`.
- **SystemAssessment > Scan Information section:** Renders Last Scan, Scan Duration, Scan Type, Components Analyzed, Vulnerabilities Found, Next Scheduled Scan. `SystemAssessment.tsx`.
- **SystemsList > Summary Stats:** Four stat tiles at bottom of Fleet Management: Total Systems, Healthy, Warning, Critical. `SystemsList.tsx`.
- **Dashboard > Recommended Actions panel:** Blue banner with top recommendation, confidence, system count, and "View All Recommendations" CTA. `Dashboard.tsx`.
- **Dashboard > Systems Requiring Attention widget:** Shows critical systems with stability score, risk level, and recommended action. h2 is "Systems Requiring Attention" (not "Systems at Risk"). `Dashboard.tsx`.
- **Dashboard > Performance Trends chart:** Line chart for Stability, Responsiveness, Network over the last 24 hours. `Dashboard.tsx`.
- **Dashboard > Active Alerts panel:** Alert list with severity, message, and timestamp. `Dashboard.tsx`.
- **Dashboard > Recent Activity panel:** Recent system events with icons and timestamps. `Dashboard.tsx`.
- **Dashboard > Systems Health Overview table:** Table of all systems with Status (Healthy, Warning, Critical), Stability Score, and Last Upgrade. `Dashboard.tsx`.
- **ModernizationAdvisor > Preview Dry Run modal:** `PreviewDryRun` component imported and toggled by `showPreview` state. Not described in the get-a-recommendation docs. `ModernizationAdvisor.tsx`.
- **OnboardingTour component:** Interactive tour UI rendered via `OnboardingTour` in `Root.tsx`. The tour trigger is documented, but the tour steps and content are not described in any guide page.
- **GlossaryModal:** Full glossary modal opened from sidebar. The glossary trigger is documented in getting-started.md, but the full glossary content is described on the Reference page. `GlossaryModal.tsx`.
- **Monitoring > Alert severity counts dashboard:** Four stat tiles at top of Monitoring page: Critical Alerts, Warnings, Info, System Health. `Monitoring.tsx`.

Summary count

GUIDE	✓ MATCHES	✗ MISMATCHES	🔍 NOT FOUND	+ UNDOCUMENTED
User Guide	68	3	3	12

GUIDE	✓ MATCHES	✗ MISMATCHES	🔍 NOT FOUND	+ UNDOCUMENTED
Admin Guide	27	1	0	6
Total	95	4	3	18

Changes since 2026-04-22

Fixed since last audit:

- Sidebar navigation labels corrected throughout (previous audit found mismatches on "Overview" vs "Dashboard", "Systems" link destination, and several page h1 discrepancies).
- Demo Mode active state label corrected: "● Demo Mode Active" now matches the prototype.
- Register a System modal: all step labels, field names, and button labels corrected to match prototype.
- Modernization page: page correctly identified as "Upgrade Profiles" (h1) with sidebar link "Modernization."
- Simulation button label corrected to "Run it for me."
- Simulation check names corrected: "Disk space available", "Backup verification", "Quirk conflicts."
- Apply button labels corrected: "Apply Stable Profile", "Apply Performance Profile", "Apply Extended Operation Profile."
- Deployment phase names corrected to match prototype: "Backup & Validation", "Executing Upgrade", "Testing & Verification."
- Rollback button label corrected: "Roll back deployment."
- Monitoring page title corrected: "Monitoring & Alerts."
- Monitoring confirmed as showing past 24 hours of data with no time-range selector, resolving the conflict between monitor.md and monitor-system-health.md.
- Troubleshooting page fully rewritten: Active Issues, Suggested Fixes, System Logs, Diagnostic Tools all now match prototype structure and labels.
- Automation page: Create Pipeline button location, Pipeline Visualization stage names, Scheduled Jobs, Event Hooks, and Recent Pipeline Runs all corrected.
- Audit Logs filter options corrected to match prototype dropdown values.
- Admin guide tabs corrected: Users & Access, General, Notifications.
- Developer Hub API Keys tab coverage added with accurate scope list and column names.

- Connect Integrations rewritten to remove planning-note framing and correctly identify integration categories.

Remaining issues:

- `getting-started.md` : Demo Mode inactive button label described as "Demo Mode button" but prototype renders "Try demo — your real data stays safe." `Root.tsx` line 115.
- `assess-your-system.md` : Dependency graph node click described as opening a detail panel. No `onClick` handler exists on graph nodes. `SystemAssessment.tsx` .
- `assess-your-system.md` : Risk table column headers described as sortable. No sort logic in `SystemAssessment.tsx` .
- `assess-your-system.md` : Assessment gate described as conditionally disabling the Proceed to Modernization button. The button is always an active `<Link>` with no conditional disable. `SystemAssessment.tsx` .

Report generated 2026-05-09 by static analysis of source.

Tags: `nimbuswiz` `qa`