



TraceBoard Suite User Manual

Version 1.0.6 | July 2026

Requirements Management · Compliance Documents · Test Execution

Generated on July 30, 2026

TraceBoard Suite — Professional Traceability Platform

Table of Contents

1 Introduction

- 1.1 What is the TraceBoard Suite?
- 1.2 Who Is It For?

2 Getting Started

- 2.1 System Requirements
- 2.2 Installation
- 2.3 Licensing
- 2.4 First Login

3 TraceBoard – Core Traceability

- 3.1 Projects and Sprints
- 3.2 Items – Requirements, Tasks, Bugs, Tests
- 3.3 Relations and Traceability
- 3.4 Trace Views
- 3.5 Baselines
- 3.6 Import / Export
- 3.7 LLM Chat
- 3.8 Generate Test Cases
- 3.9 Cross-Module Consistency Check
- 3.10 Audit Narrative
- 3.11 Import Assistant

4 TraceDocs – Document Generation

- 4.1 Overview
- 4.2 Document Management
- 4.3 Generating Documents
- 4.4 Document Templates
- 4.5 Markdown Editor
- 4.6 Export Word

5 TraceTest – Test Execution and Coverage

- 5.1 Overview
- 5.2 Test Cases
- 5.3 Test Runs
- 5.4 Coverage Report
- 5.5 Evidence Upload
- 5.6 Coverage Trends
- 5.7 Test Report Export

6 Git Integration

- 6.1 Key Capabilities

6.2 Quick Start

6.3 Where Commits Appear

7 Product Switcher

8 Administration

8.1 User Management

8.2 SSO (OIDC)

8.3 LLM Configuration

8.4 Backup and Restore

8.5 Onboarding Tour

9 Troubleshooting

9.1 License Not Found or Expired

9.2 TraceDocs or TraceTest Cannot Connect to TraceBoard

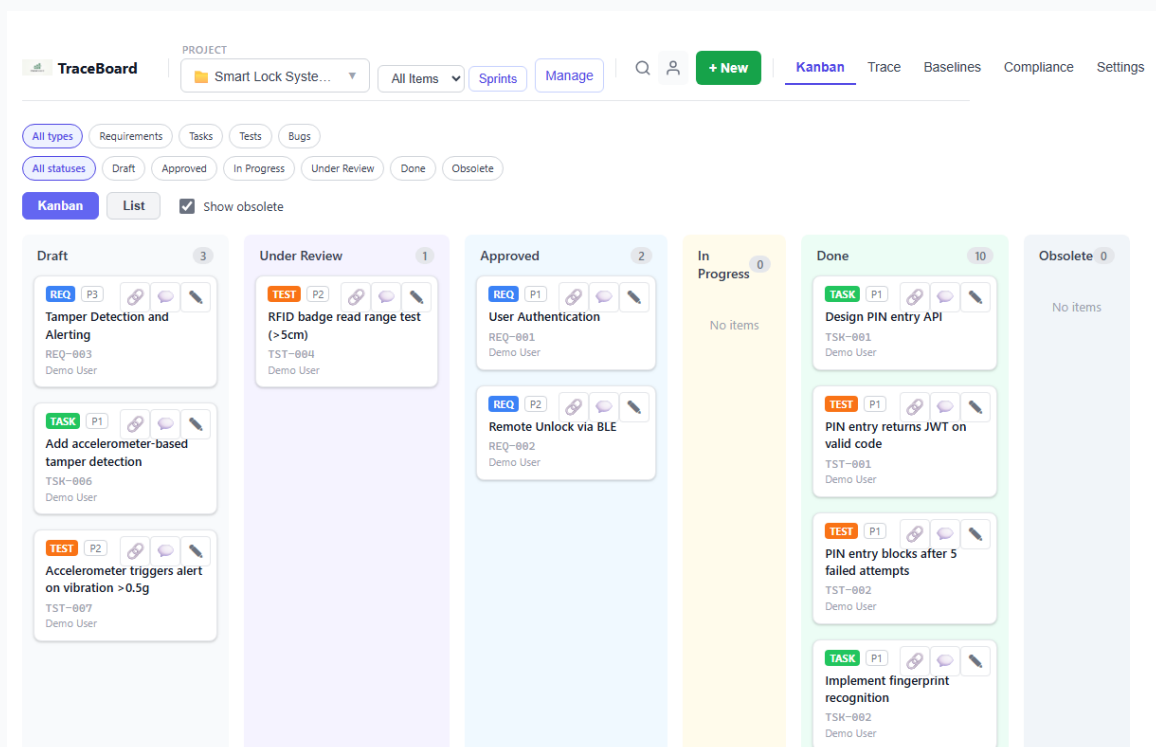
9.3 LLM (AI) Connection Issues

9.4 PDF Generation Fails

9.5 Cannot Switch Products

10 Glossary

11 Support



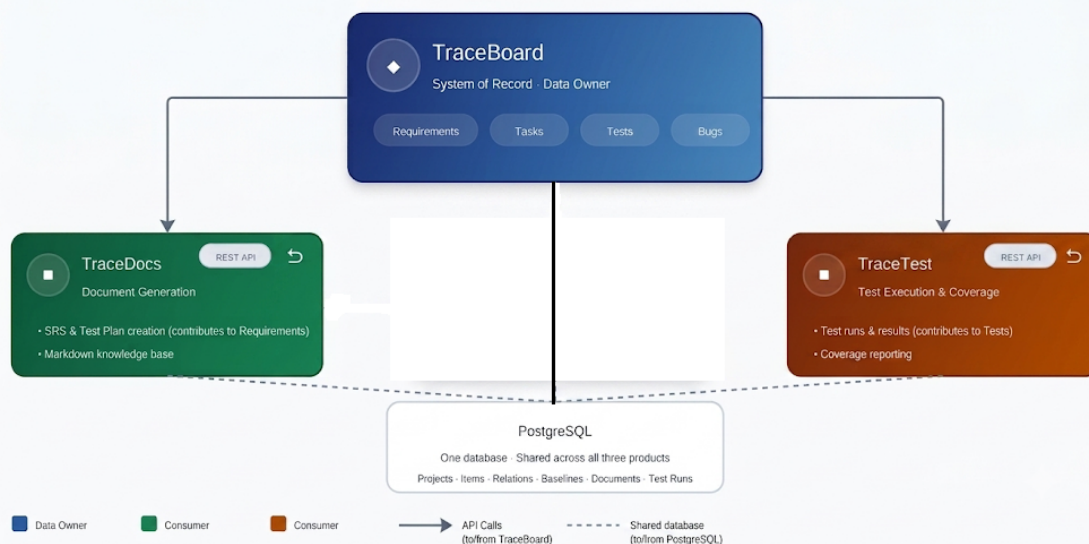
1 Introduction

1.1 What is the TraceBoard Suite?

The TraceBoard Suite is an integrated platform for requirements management, compliance document generation, and test execution tracking. It consists of three products that work together from a single source of truth:

Product	Purpose
TraceBoard	Requirements, baselines, traceability, and test planning. The core product — owns all domain data.
TraceDocs	Compliance document generation (SRS, Test Plans) directly from TraceBoard baselines.
TraceTest	Test execution tracking, evidence management, and coverage reporting.

All three products share a common interface, a single PostgreSQL database, and the same authentication system. TraceBoard is the sole owner of projects, baselines, requirements, test cases, and traceability links. TraceDocs and TraceTest consume that data but never modify it — they store only their own operational data (document generations, test results, evidence).



1.2 Who Is It For?

- **Engineering teams** who need structured requirements management and traceability.
- **Compliance officers** who need audit-ready documentation (SRS, Test Plans) with verified traceability.
- **Quality assurance teams** who need to track test execution and measure coverage against requirements.
- **Project managers** who need to oversee deliverables and review compliance status.

2 Getting Started

2.1 System Requirements

- **Docker** and **Docker Compose** (latest stable versions recommended).
- Recommended hardware: 4 GB RAM, 2 CPU cores, 10 GB free disk space.
- Optional: Ollama if you want to use the LLM chat feature (see Section 7.3).

2.2 Installation

1. Download the `docker-compose.yml` file from the project repository.
2. Obtain a license file (`traceboard.license`) — see Section 2.3.
3. Place the license file in the same directory as `docker-compose.yml`.
4. Open a terminal in that directory and run:

```
bash
```

```
docker compose up -d
```

```
^
```

Docker pulls the required images and starts all services: PostgreSQL, TraceBoard backend, TraceDocs backend, and TraceTest backend.

1. Once the services are healthy, access the suite at:

Product	URL
TraceBoard	http://localhost:8000
TraceDocs	http://localhost:8001
TraceTest	http://localhost:8002

```
root@demo:/opt/traceboard-suite# docker compose up -d
[+] Running 6/6
✔Network traceboard-suite_suite-net          Created           0.1s
✔Container traceboard-suite-db-1             Healthy           7.2s
✔Container traceboard-suite-traceboard-backend-1 Healthy         12.0s
✔Container traceboard-suite-tracetest-backend-1 Started          12.4s
✔Container traceboard-suite-tracedocs-backend-1 Started          12.3s
✔Container traceboard-suite-nginx-1          Started          12.6s
root@demo:/opt/traceboard-suite#
```

2.3 Licensing

The TraceBoard Suite uses an RSA-signed JSON license file to control access to product features.

How to obtain a license:

- Contact your sales representative for a trial or full license.
- You will receive a file named `traceboard.license`.

How to install the license:

- Place `traceboard.license` in the same directory as your `docker-compose.yml`.
- The file is automatically picked up by all services when they start.

License file format:

The license is a JSON document containing:

- `subject`: the licensed organization or individual.

- `email`: contact email.
- `issued_at` and `expires_at`: validity period.
- `products`: an array listing which products are entitled (e.g., `["TraceBoard", "TraceDocs", "TraceTest"]`).
- `signature`: a cryptographic RSA signature that prevents tampering.

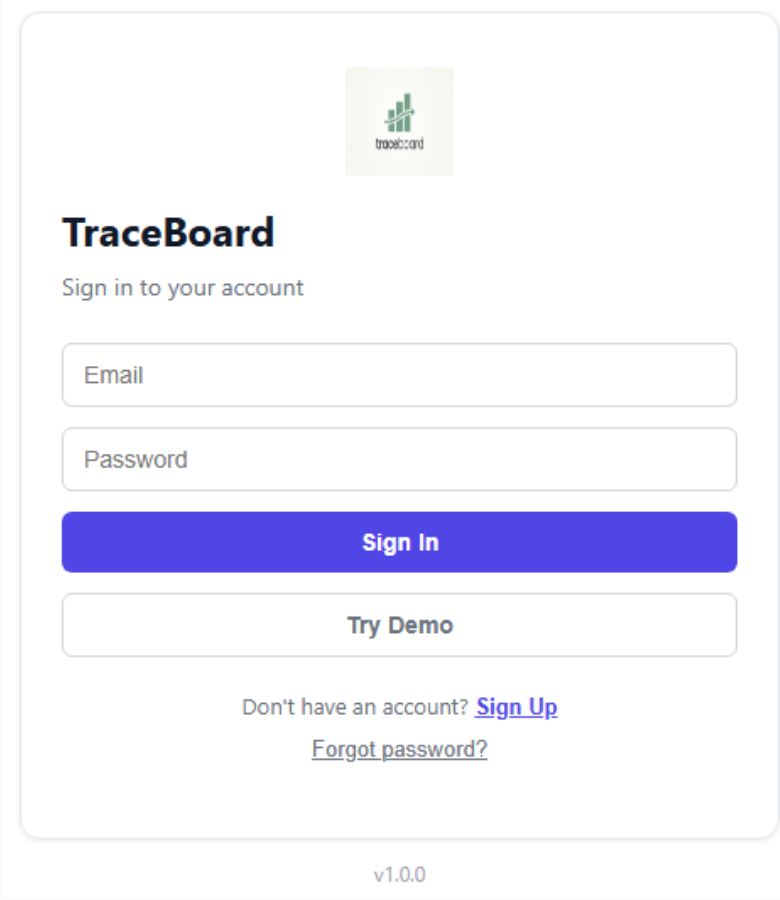
What happens if the license is missing or expired?

- If no license file is found, the services start in degraded mode or refuse to start (depending on configuration).
- If the license has expired, the products show a warning banner and restrict access to licensed features.
- Contact your administrator or sales representative to renew or obtain a new license.

2.4 First Login

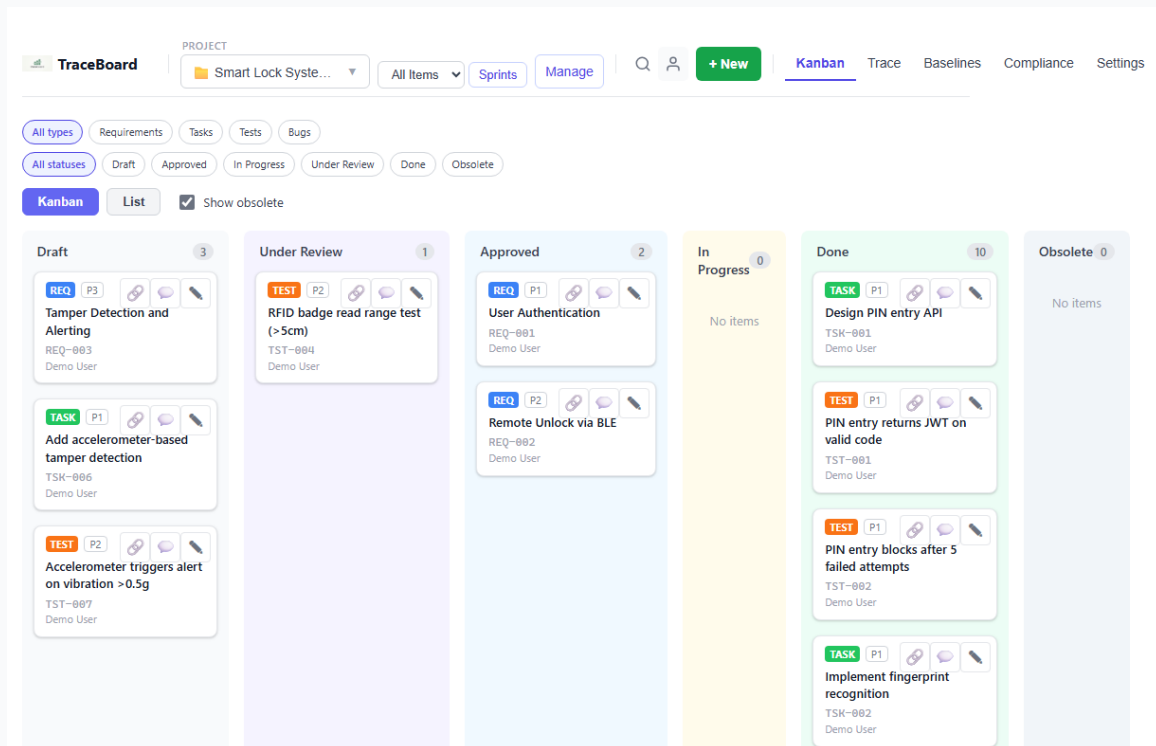
1. Open TraceBoard at `http://localhost:8000`.
2. Log in with the default credentials:
 1. **Change the password immediately** from the user settings page.
 2. (Optional) A demo login may be available for evaluation — check with your administrator.

All three products share the same authentication session. Logging into one product logs you into all three.

The image shows the TraceBoard login page. At the top center is the TraceBoard logo, which consists of a green bar chart icon above the word "traceboard". Below the logo, the word "TraceBoard" is displayed in a large, bold, black font. Underneath that, the text "Sign in to your account" is shown in a smaller, gray font. There are two input fields: "Email" and "Password", both with light gray borders and placeholder text. Below these fields is a prominent blue button with the text "Sign In" in white. Underneath the blue button is a white button with a gray border and the text "Try Demo" in gray. At the bottom of the form, there is a link "Don't have an account? Sign Up" and another link "Forgot password?". The version number "v1.0.0" is displayed at the very bottom of the page.

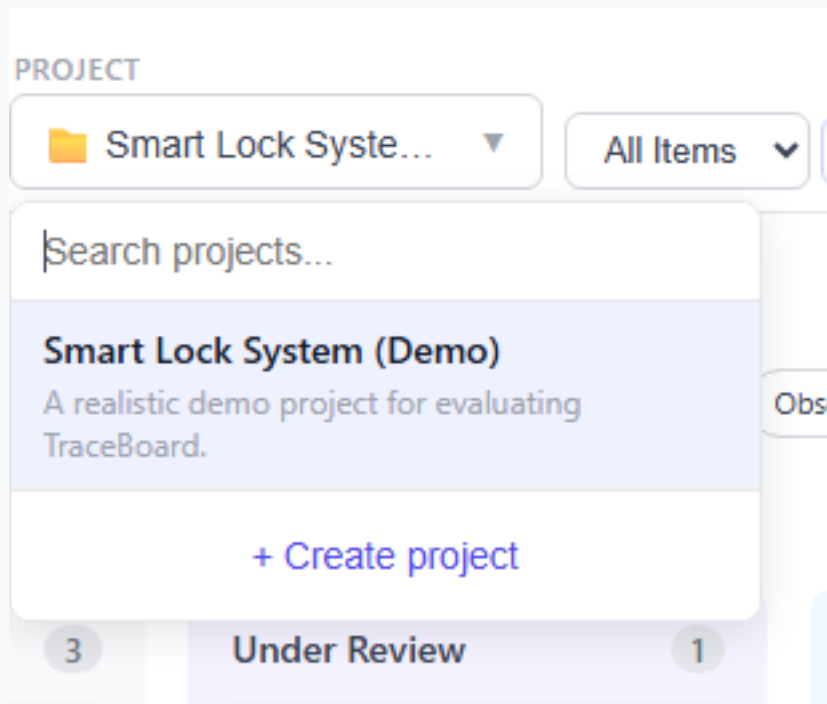
3 TraceBoard – Core Traceability

TraceBoard is the core product of the suite. It owns all domain data: projects, requirements, test cases, traceability links, and baselines. Everything in TraceDocs and TraceTest originates from TraceBoard.



3.1 Projects and Sprints

- **Projects** organize your work into separate units. Each project has its own requirements, test cases, baselines, and settings.
- Create a project from the project selector dropdown in the top navigation bar.
- Within a project, you can define **Sprints** to group work into time-boxed iterations.
- Switch between projects using the project selector — all views update to show the selected project's data.



3.2 Items – Requirements, Tasks, Bugs, Tests

Items are the building blocks of traceability in TraceBoard. Each item belongs to a project and has a type:

Type	Purpose
Requirement	A functional or non-functional requirement (e.g., "The system shall allow users to reset their password").
Task	A unit of development work.
Bug	A defect report.
Test	A test case definition (used by TraceTest for execution).

Item properties:

- **Status:** Draft, In Progress, Review, Approved, Closed, etc.
- **Priority:** Low, Medium, High, Critical.
- **Owner:** The user responsible for the item.
- **Title and Description:** Free-text with Markdown support.
- **External ID:** A reference ID from an external system (e.g., JIRA, Azure DevOps).

Creating items:

1. Select a project from the project selector.
2. Click **New Item** (or the + button).
3. Choose the item type, fill in the details, and click **Create**.

Editing and deleting:

- Click an item to open its detail view. Edit fields inline or in a form.
- Delete an item via the actions menu (three dots). Deletion removes the item and all its relations. This action cannot be undone.

REQUIREMENT Remote Unlock via BLE
3668e25d-f45e-48e8-9248-5823181a5d51

[Edit](#) [Comments](#) [History](#) [Attachments](#) [Links](#)

Type: requirement Version: 1 Short ID: REQ-002 [View Trace](#)

Title
Remote Unlock via BLE

Description [Preview](#)
The system must allow remote unlocking via Bluetooth Low Energy from an authorised mobile app.

Status
Approved

Priority (1-5)
2

Owner
Demo User

Sprint
None (Backlog)

Estimated hours
e.g., 8

3.3 Relations and Traceability

Relations link items together to create a traceability graph. Each relation has a type:

Relation Type	Meaning
implements	A task implements a requirement.
verifies	A test verifies a requirement.
parent	Parent-child hierarchy between items (e.g., a requirement decomposed into sub-requirements).
relates-to	A generic bidirectional link.

Creating relations:

1. Open an item's detail view.
2. In the **Relations** section, click **Add Relation**.
3. Select the target item and the relation type.
4. Click **Create**.

The trace tree:

- The trace tree shows an item and all its recursively linked items.
- Expand/collapse branches to explore the trace graph.
- The tree automatically detects and flags **circular references** (e.g., $A \rightarrow B \rightarrow C \rightarrow A$). Circular references are highlighted in red with a warning icon.

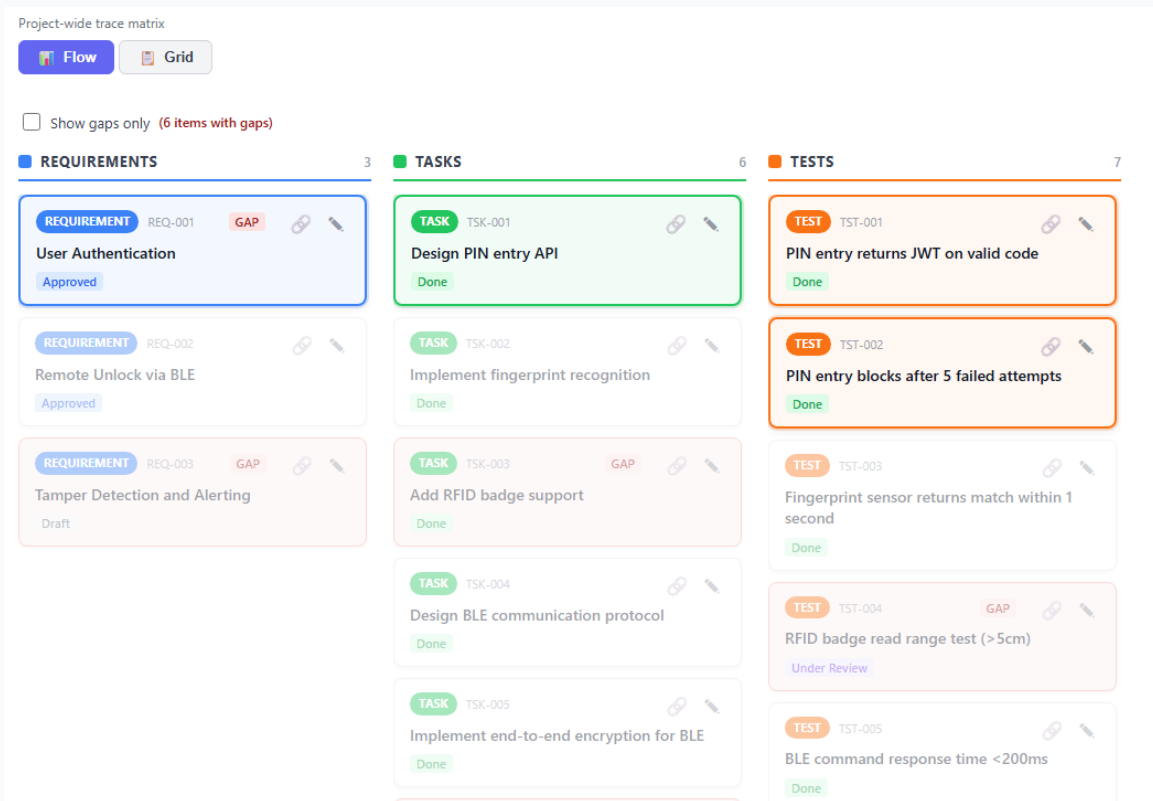
Tip: Use `verifies` relations to build the compliance matrix between requirements and tests. This matrix is used by TraceTest for coverage reporting.

3.4 Trace Views

TraceBoard provides multiple ways to visualize traceability:

Flow View:

- An interactive visual graph showing items as nodes and relations as directed edges.
- Pan, zoom, and drag nodes to explore the graph.
- Click a node to open the item detail.
- Different relation types are shown with different colored arrows.



Grid View (Compliance Matrix):

- A table showing requirements (rows) against tests (columns).
- Each cell shows whether a `verifies` relation exists.
- Use this view to quickly see which requirements lack test coverage.

Project-wide trace matrix

Flow Grid

3 requirements · 6 tasks · 7 tests · 2 gaps detected

☐ Show gaps only

	REQUIREMENT ID	REQUIREMENT TITLE	LINKED TASK ID	LINKED TASK TITLE	LINKED TEST ID	LINKED TEST TITLE	TEST STATUS
 	REQ-001	User Authentication	TSK-001	Design PIN entry API	TST-001	PIN entry returns JWT on valid code	Passed
 	REQ-001	User Authentication	TSK-001	Design PIN entry API	TST-002	PIN entry blocks after 5 failed attempts	Passed
 	REQ-001	User Authentication	TSK-002	Implement fingerprint recognition	TST-003	Fingerprint sensor returns match within 1 second	Passed
 	REQ-001	User Authentication	TSK-003	Add RFID badge support	TST-004	RFID badge read range test (>5cm)	Review
 	REQ-002	Remote Unlock via BLE	TSK-004	Design BLE communication protocol	TST-005	BLE command response time <200ms	Passed
 	REQ-002	Remote Unlock via BLE	TSK-005	Implement end-to-end encryption for BLE	TST-006	Encrypted payload cannot be replayed	Passed
 	REQ-003	Tamper Detection and Alerting	TSK-006	Add accelerometer-based tamper detection	TST-007	Accelerometer triggers alert on vibration >0.5g	Draft

 Highlighted rows = missing coverage (gap)

Gap Detection:

- Toggle the **"Show Gaps only"** filter to display only requirements that have no linked tests.
- This highlights coverage gaps in your trace model.

3.5 Baselines

A baseline is a point-in-time snapshot of all items and relations in a project. Baselines are used by TraceDocs to generate documents and by TraceTest to create test runs. Once created, a baseline is immutable — it captures the exact state of your project at that moment.

Creating a baseline:

1. Navigate to the **Baselines** tab in TraceBoard.
2. Click **New Baseline**.
3. Provide a name and an optional tag (e.g., v1.0, Sprint-3).
4. Click **Create**. All items and relations in the current project are frozen in the snapshot.

Comparing baselines:

1. In the Baselines view, select two baselines using the checkboxes.
2. Click **Compare**.
3. The comparison view shows:

Baselines

Create Baseline

Baseline created successfully X

Name	Created	ID
Test Readiness Review	04/07/2026, 19:36:00	19e18308...
Initial Design Review	04/07/2026, 19:24:17	7fab7835...

Compare Baselines

Baseline A

Initial Design Review

Baseline B

Test Readiness Review

Compare

☐ Show system fields

Modified Items (2)

3f221966

Field	Old Value	New Value
status	under_review	done

61a2f8f8

Field	Old Value	New Value
status	draft	approved

3.6 Import / Export

Import from Excel:

1. Go to the project's import page.
2. Upload an .xlsx file with your items.
3. The Excel file must include columns for type, title, description, status, priority, and owner.
4. **Important:** Relations must use item UUIDs (not titles). The Excel file needs a relations sheet with source_uuid, target_uuid, and relation_type columns.
5. Review the preview and click **Import**.

Export to Excel:

1. Go to the project's export page.
2. Click **Export Excel**.
3. The downloaded file includes an items sheet (all items with their UUIDs) and a relations sheet (all relations with source/target UUIDs).
4. This file can be re-imported (e.g., for bulk editing or backup).

3.7 LLM Chat

TraceBoard includes an "Ask AI" assistant that can answer questions about your trace data using natural language. This feature requires an LLM endpoint to be configured — either a local Ollama instance or a cloud provider (see Section 7.3).

Enabling the feature:

- Once the LLM is configured with LLM_API_URL and LLM_MODEL environment variables, an "Ask AI" button appears in TraceBoard.

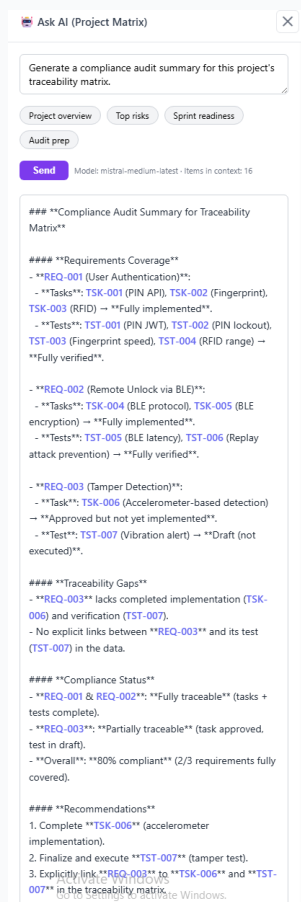
Using the Ask AI feature:

1. Click the **"Ask AI"** button in the TraceBoard toolbar.
2. Type your question in natural language (e.g., "Which requirements have no linked tests?").
3. The AI responds with a detailed answer. **Citations** appear as clickable references — click a citation to navigate directly to the referenced item.
4. You can ask follow-up questions to drill deeper into the data.

Example questions you can ask:

- "Which requirements have no linked tests?" (gap analysis)
- "Summarise the trace coverage for project X."
- "What is the highest priority requirement that has not been verified?"
- "Identify any risks based on the current trace data."

The AI assistant has read-only access to your project's items, relations, and baselines. It does not modify any data.



3.8 Generate Test Cases

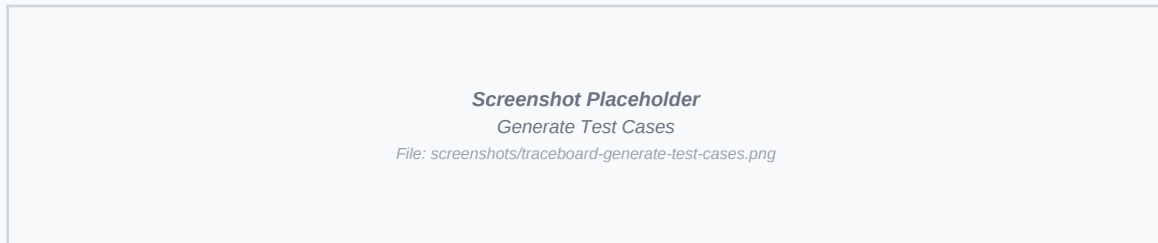
TraceBoard can automatically generate test cases from your requirements and tasks using AI. This feature uses the configured LLM to analyse your items and propose relevant test cases.

Generating test cases:

1. Navigate to a project and open the **Generate Test Cases** dialog (found in the project actions menu).
2. Select which requirements or tasks you want to generate test cases from.
3. Click **Generate**. The AI analyses the selected items and proposes test cases.

4. Review the generated test cases in the preview table. Each proposed test case includes a title, description, and suggested priority.
5. **Edit** any test case before importing by clicking on it in the preview.
6. **Deselect** any test cases you do not want to import.
7. Click **Import** to create the selected test cases in your project. New test cases are automatically linked to their source items via *verifies* relations.
8. Refresh the item list to see the newly created test cases.

Tip: You can re-run the generation with different selections or LLM settings. Generated test cases are only created after you explicitly confirm the import.



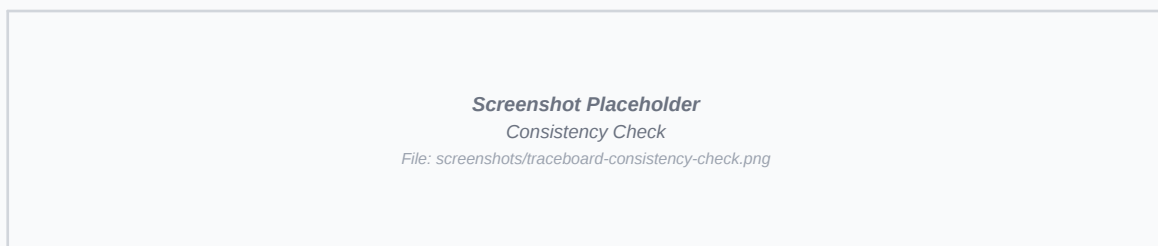
3.9 Cross-Module Consistency Check

The consistency check detects contradictions and inconsistencies across your requirements, design documents, and test cases. This is especially valuable for compliance with standards like ISO 26262 and IEC 62304.

Running a consistency check:

1. Navigate to the **Consistency** tab in TraceBoard (or access it from the AI tools menu).
2. Select the scope of the check — you can run it against a specific baseline, a project, or across multiple modules.
3. Click **Run Check**. The AI analyses your data and returns a list of findings.
4. Each finding includes:
 1. Review the findings and address them by updating items, adding missing relations, or creating new test cases.

Tip: Run consistency checks before creating baselines and generating compliance documents to catch issues early.



3.10 Audit Narrative

The audit narrative feature generates a compliance-ready narrative document from your project data. It produces a structured report suitable for submission to auditors, covering ISO 26262, IEC 62304, and DO-178C standards.

Generating an audit narrative:

1. Open the **Audit Narrative** tool from the project actions menu.
2. Select a baseline to use as the source of truth.
3. Choose the compliance standard (ISO 26262, IEC 62304, or DO-178C).
4. Click **Generate**. The AI produces a structured narrative with the following sections:
 1. Review the narrative in the preview pane. You can edit any section before exporting.
 2. Click **Export** to download the narrative as a PDF or copy it to your clipboard.

Tip: Use the audit narrative together with TraceDocs document generation for a complete compliance package.

Screenshot Placeholder
Audit Narrative
File: screenshots/traceboard-audit-narrative.png

3.11 Import Assistant

The Import Assistant extracts structured requirements from Word (.docx) and PDF documents using AI. Instead of manually retyping requirements from specification documents, you can upload them and let the AI do the extraction.

Using the Import Assistant:

1. Navigate to the **Import Assistant** from the project import page or the AI tools menu.
2. Upload a Word (.docx) or PDF document containing requirements (e.g., a customer specification, regulatory document, or legacy SRS).
3. Click **Analyse**. The AI processes the document and extracts candidate requirements.
4. Review the extracted items in the preview table. Each row shows the proposed title, description, type, and priority.
5. **Edit** any item by clicking on it — adjust the title, description, type, or priority as needed.
6. **Deselect** items you do not want to import.
7. Click **Import** to create the selected items in your project.
8. After import, you can add relations, assign owners, and include the items in baselines as usual.

Supported formats:

- **Word (.docx)** — Standard Word documents.
- **PDF** — Text-based PDF documents (scanned documents without OCR may not extract well).

Tip: The Import Assistant works best with structured documents where requirements are clearly labelled (e.g., "REQ-001", "The system shall...").

Screenshot Placeholder
Import Assistant
File: screenshots/traceboard-import-assistant.png

4 TraceDocs – Document Generation

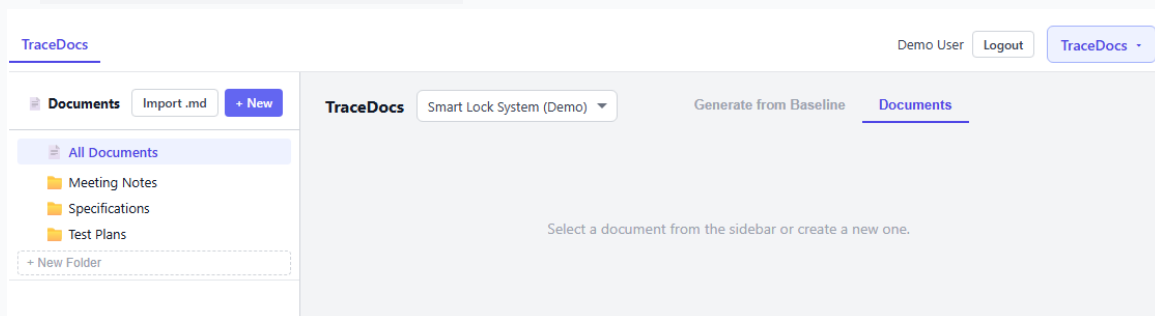
4.1 Overview

TraceDocs generates compliance-grade documents — Software Requirements Specifications (SRS) and Test Plans — directly from TraceBoard baselines. Instead of manually copying and formatting data, you select a baseline, choose a template, and TraceDocs produces a professional PDF or Word document. Every generated document embeds its source baseline ID for full traceability.

4.2 Document Management

Creating a document generation:

1. Navigate to TraceDocs (use the Product Switcher or go to `http://localhost:8001`).
2. Click **New Document**.
3. Configure your document:
 1. Click **Generate**. A progress indicator shows the generation status.
 2. When complete, review the document in the **HTML Preview**.
 3. Click **Export** and choose **PDF** or **Word** format to download. The file is downloaded with a descriptive filename: `ProjectName_DocType_BaselineTag_Date.pdf` (e.g., `MyProject_SRS_v1.0_2026-06-29.pdf`).

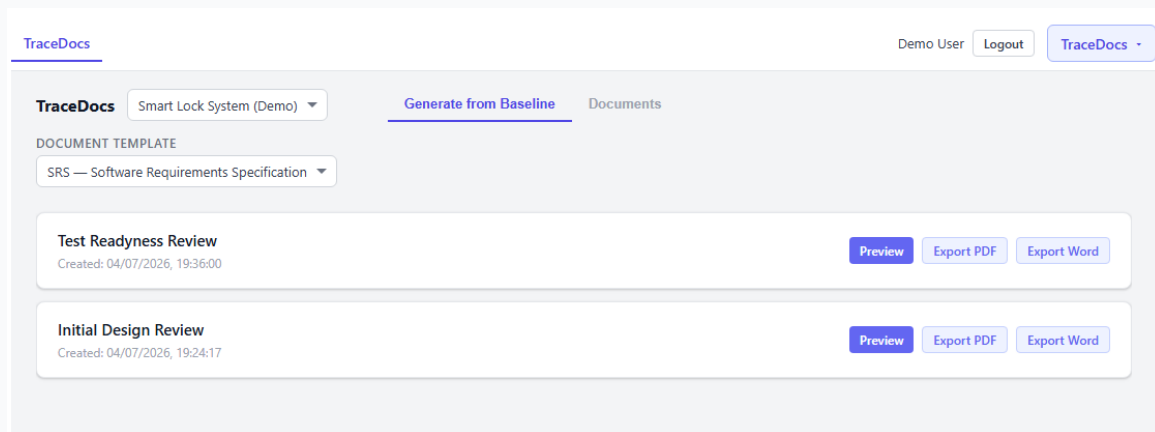


Organising documents into folders (future feature):

- Documents can be organised into a nested folder hierarchy.
- Create, rename, and delete folders.
- Drag and drop documents to move them between folders.

Generation history:

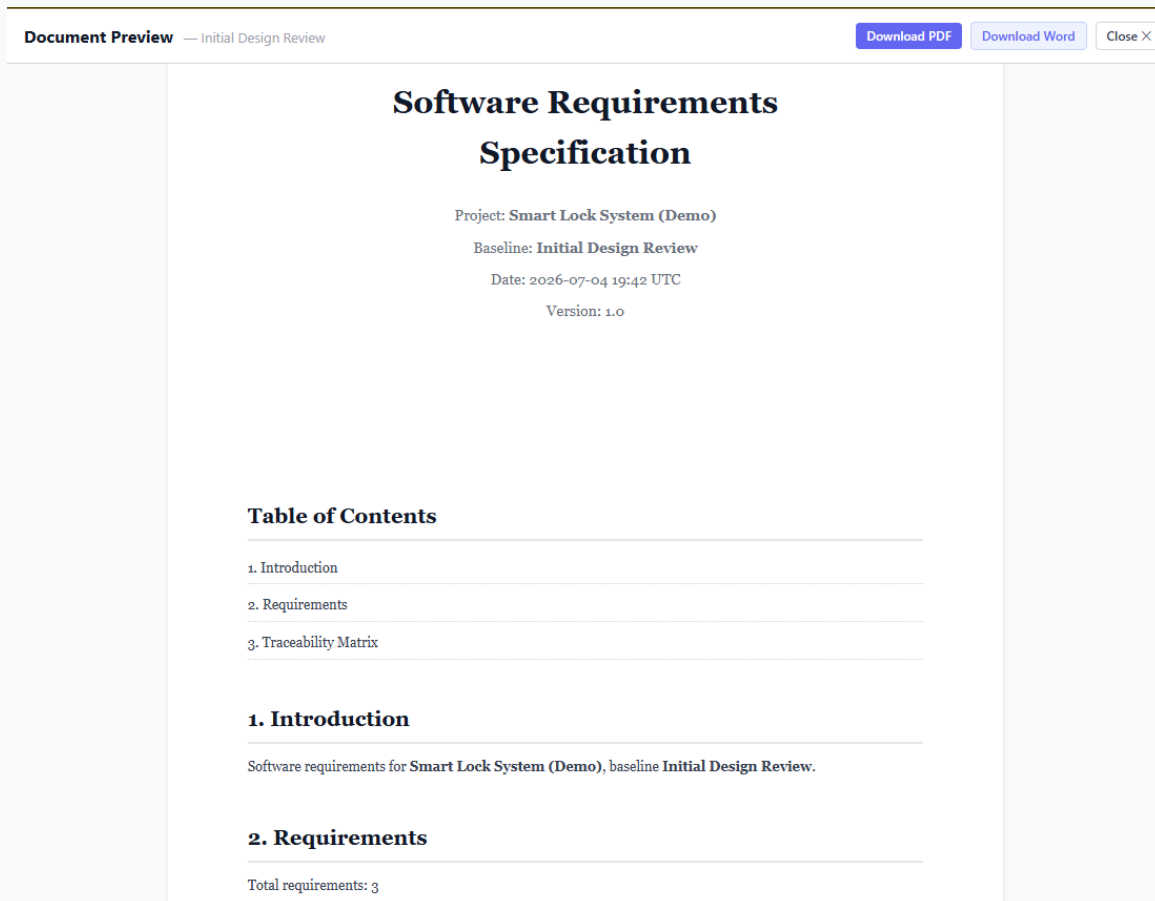
1. Click the **History** tab in TraceDocs.
2. View a list of all past document generations, including date, template type, baseline used, and status.
3. Click **Download** to re-download a previously generated file.
4. Click **Re-generate** to pre-fill a new document form with the same settings.



4.3 Generating Documents

To generate a document:

1. **Select a baseline** — this determines which requirements and test cases are included.
2. **Choose a template** — SRS or Test Plan (see Section 4.4).
3. **Click "Generate Document"** — the backend fetches the baseline data from TraceBoard, merges it with the template, and produces a styled document.
4. **Preview** — review the HTML preview in your browser.



1. **Export** — download as PDF or Word (.docx). The filename is descriptive:
ProjectName_DocType_Baseline_Date.pdf.

The PDF includes a cover page, auto-generated table of contents, styled headings, page numbers, and a generation timestamp. The Word export mirrors the PDF styling with a Word-native TOC field.

[Preview](#)[Export PDF](#)[Export Word](#)

4.4 Document Templates

TraceDocs comes with two built-in templates:

Template	Description
SRS	Software Requirements Specification — structured with sections for introduction, functional requirements, non-functional requirements, interfaces, and appendices.
Test Plan	Test planning document — includes test strategy, scope, test cases table, schedule, and resource sections.

Templates currently planned for future releases:

- **Design Description** — software architecture and component design.
- **User Manual** — end-user documentation.
- **Custom templates** — user-defined template creation via the UI.

4.5 Markdown Editor

TraceDocs includes a built-in Markdown editor for writing knowledge base articles, design notes, and supplementary documentation.

Using the editor:

1. Navigate to the **Knowledge Base** section in TraceDocs.
2. Click **New Article** to create a new document.
3. Write your content in Markdown in the left pane. The right pane shows a **live preview** of the rendered output.
4. Organise articles into **folders** — create a folder structure that matches your project's documentation needs.
5. Click **Save** to persist your article.

Tip: Use the Markdown editor to document design decisions, meeting notes, or supplementary material that complements your generated SRS and Test Plan documents.

4.6 Export Word

Generated documents can be exported as Word (.docx) files for further editing in Microsoft Word or compatible editors.

Exporting to Word:

1. After generating a document, click the **Export** button in the preview toolbar.
2. Select **Word (.docx)** from the dropdown menu.
3. The file downloads with a descriptive filename: `ProjectName_DocType_BaselineTag_Date.docx` (e.g., `MyProject_SRS_v1.0_2026-06-29.docx`).

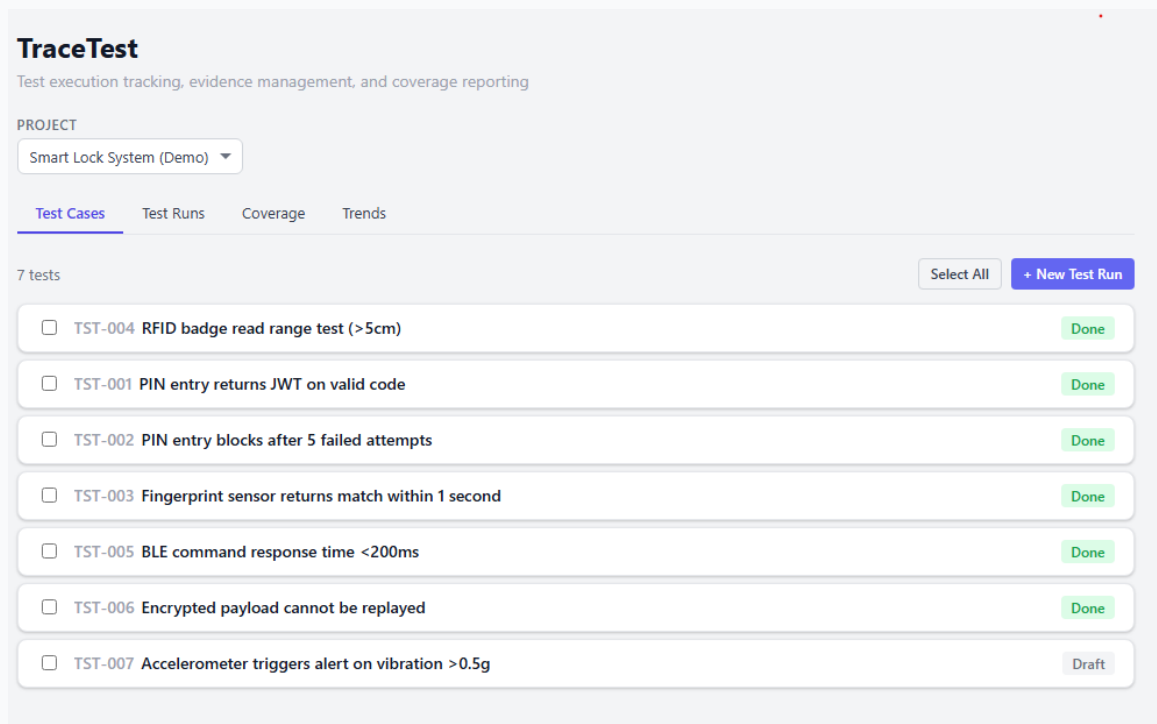
4. Open the file in Microsoft Word or your preferred editor. The document includes:

Tip: The Word export is ideal for teams that need to add final polish, signatures, or custom formatting before submission.

5 TraceTest – Test Execution and Coverage

5.1 Overview

TraceTest tracks the execution of test cases defined in TraceBoard baselines. Testers record results (Pass/Fail/Blocked/Not Run), attach evidence, and generate coverage reports that show which requirements are verified by passing tests.



5.2 Test Cases

All items of type test from TraceBoard automatically appear as test cases in TraceTest. Test cases are grouped by project and are read-only — they are managed in TraceBoard.

To view test cases:

1. Navigate to TraceTest (use the Product Switcher or go to `http://localhost:8002`).
2. Select a project from the project selector.
3. The test cases for that project are listed, grouped by the baseline they belong to.

5.3 Test Runs

A **Test Run** is a named execution session where test cases from a baseline are executed and results are recorded.

Creating a test run:

1. In TraceTest, click **New Test Run**.
2. Provide a name for the test run.

3. Select a baseline from the dropdown (populated from TraceBoard projects).
4. Optionally, select specific test cases to include — or include all test cases from the baseline.
5. Click **Create**. TraceTest seeds the run with the selected test cases, all initially set to "Not Run".

×

New Test Run

Run Name

All 7 tests will be included

CancelCreate Run

Recording results:

1. Open a test run from the runs list.
2. For each test case, click the status chip to set the result:
 1. Changes save automatically — no save button needed.
 2. Optionally, add a **note** to each result explaining the outcome or linking to a bug report.

TraceTest

Test execution tracking, evidence management, and coverage reporting

PROJECT

Smart Lock System (Demo) ▾

Test Cases **Test Runs** Coverage Trends

1 test run

[+ New Test Run](#)**Sprint 3 Regression**

7 tests, 0 pass, 0 fail, 0 blocked, 04/07/2026

Close

Delete

TST-004 RFID badge read range test (>5cm)

Pass

Fail

Blocked

Upload evidence

TST-001 PIN entry returns JWT on valid code

Pass

Fail

Blocked

Upload evidence

TST-002 PIN entry blocks after 5 failed attempts

Pass

Fail

Blocked

Upload evidence

TST-003 Fingerprint sensor returns match within 1 second

Pass

Fail

Blocked

Bulk updates:

1. Use checkboxes to select multiple test cases.
2. Click **Bulk Update**.
3. Choose the new status to apply to all selected test cases.
4. Confirm the action. All selected test cases are updated simultaneously.

Filtering and search:

- Click status chips (Pass, Fail, Blocked, Not Run) to filter the list.
- Use the free-text search to find test cases by title or external ID.
- Filter by functional area via the dropdown.
- All filter selections are preserved in the URL for sharing and bookmarking.

5.4 Coverage Report

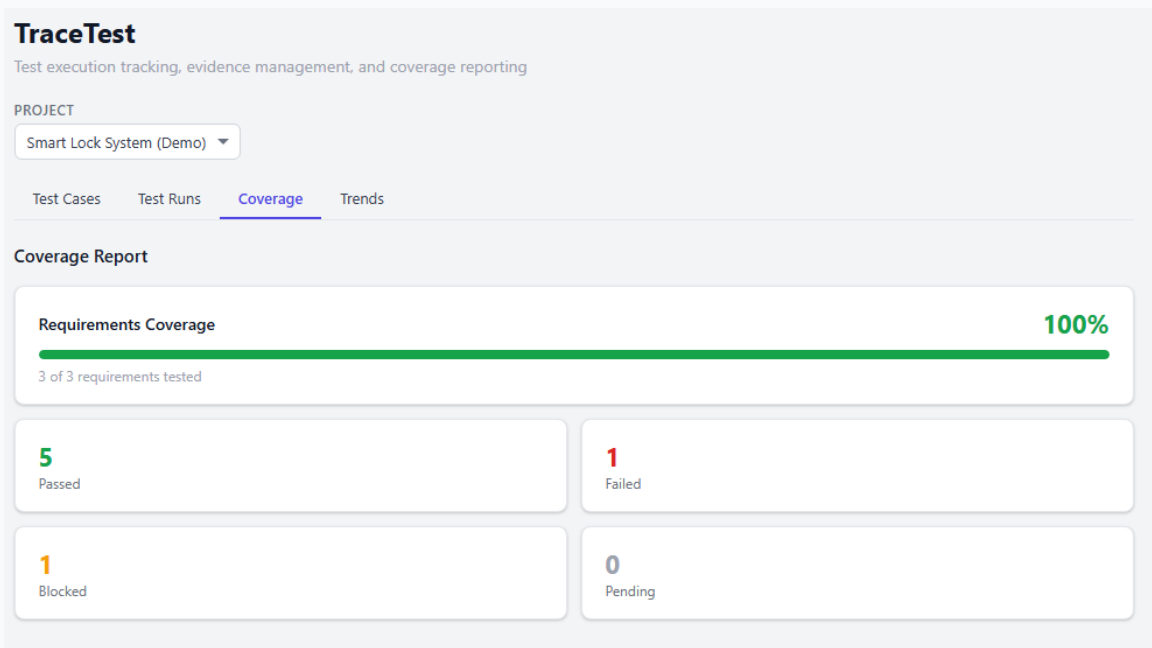
Coverage is calculated automatically based on `verifies` relations from TraceBoard and test results from TraceTest.

How coverage is calculated:

- A requirement is **Covered** if at least one linked test case has a passing result.
- A requirement is **Partially Covered** if some linked tests passed but others failed or are not run.
- A requirement is **Not Covered** if no linked tests have a passing result.
- **Coverage percentage** = (Covered requirements / Total requirements) × 100.

Viewing the coverage report:

1. In a test run, click the **Coverage** tab.
2. View the summary cards showing Covered, Partially Covered, Not Covered counts and the coverage percentage.
3. The requirement-level table shows each requirement with its coverage status and linked tests.
4. Click a requirement to drill down and see per-test results.



5.5 Evidence Upload

Attach files (screenshots, logs, test data) to individual test results as evidence of execution.

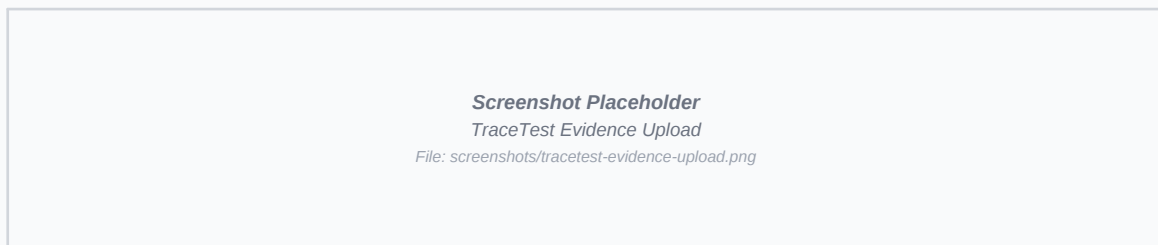
Uploading evidence:

1. Open a test run and locate the test result you want to attach evidence to.
2. Click the **Attach Evidence** button (paperclip icon) next to the result.
3. Select one or more files from your computer. Supported file types: PNG, JPG, GIF, TXT, LOG, CSV, JSON, PDF, ZIP.
4. Maximum file size: 25 MB per file.
5. Uploaded files appear in the result row with a thumbnail or file icon. Click a file to download or preview it.
6. Evidence attachments are stored alongside test results and included in exported test report PDFs.

Managing evidence:

- Click an uploaded file to download it.
- Click the delete icon to remove an attachment (requires confirmation).
- All evidence for a test run is visible in the run detail view.

Tip: Use evidence uploads to capture screenshots of test execution, log files from test runs, or signed-off test result sheets. This creates a complete audit trail for compliance reviews.



5.6 Coverage Trends

The Coverage Trends view shows how your test coverage has evolved over time across multiple test runs.

Viewing coverage trends:

1. In TraceTest, navigate to the **Coverage Trends** tab.
2. Select a project and optionally filter by time range.
3. The trends chart displays:
 1. Hover over any data point to see the exact values for that test run.
 2. Use the chart to identify coverage regressions or improvements across sprints and releases.

Interpreting the trends:

- An **upward trend** indicates increasing test coverage.
- A **downward trend** may indicate new requirements added without corresponding tests.
- **Flat lines** suggest no new testing activity.

Tip: Review coverage trends before release milestones to ensure coverage is trending in the right direction.

Screenshot Placeholder
TraceTest Coverage Trends
File: screenshots/tracestest-coverage-trends.png

5.7 Test Report Export

Export test run results and coverage data as a PDF report for distribution, archiving, or audit submission.

Exporting a test report:

1. Open a test run in TraceTest.
2. Click the **Export Report** button in the run toolbar.
3. The system generates a PDF report containing:
 1. The PDF downloads automatically. The filename includes the test run name and date:
`TestRunName_YYYY-MM-DD.pdf`.

Tip: Export reports after each test run and archive them as part of your compliance documentation. The reports embed source baseline information for full traceability.

6 Git Integration

TraceBoard Suite includes a built-in Git integration that links code commits to requirements, tasks, test cases, and documents. It supports **GitHub**, **GitLab**, and **Gitea** (including self-hosted instances) and works in air-gapped environments.

6.1 Key Capabilities

- **Automatic commit linking** — Webhooks push commit events from your repository to TraceBoard in real time.
- **Manual sync** — Fetch and process commits on demand when webhooks are not feasible.
- **Branch filtering** — Restrict processing to specific branches (e.g., `main`, `develop`).
- **Private repository support** — Access tokens are encrypted at rest.
- **Commit visibility across products** — Linked commits appear in TraceBoard item drawers, TraceDocs document views, and TraceTest test case/run details.
- **CI/CD API** — Create test runs automatically from GitHub Actions, GitLab CI, or any CI pipeline.

6.2 Quick Start

1. Navigate to your project's **Settings** → **Git Integration**.
2. Enter the **Repository URL** and select the **Git Provider**.
3. (Optional) Configure **Branch Filtering** and enter an **Access Token** for private repos.
4. Click **Save**, then **Test Connection**.
5. Set up a **webhook** in your Git provider using the URL and secret shown in the settings.

For detailed instructions, including webhook configuration, manual sync, CI/CD integration, and troubleshooting, see the Git Integration & CI/CD Guide.

6.3 Where Commits Appear

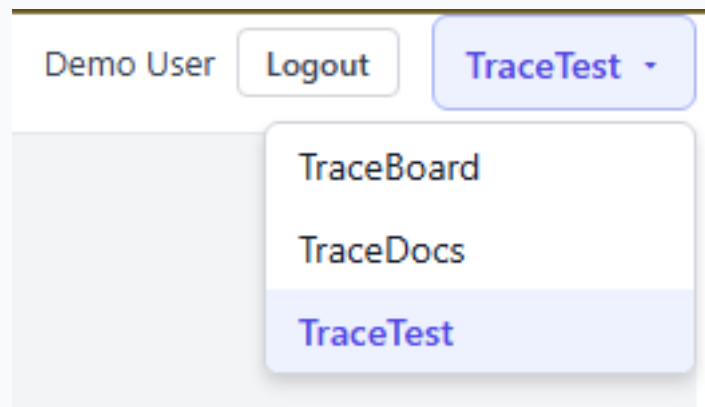
Product	Location	Details
TraceBoard	Item Cards, Trace View / Item Drawer (Commits tab), Project Commit History (sidebar)	Badge shows linked commit count; full list with SHA, message, author, date
TraceDocs	Document detail page (Commits tab)	Commits linked to items in the document's baseline or referenced in Markdown
TraceTest	Test Case Detail (Commits tab), Test Run Detail (Commits tab)	Commits linked to the test case or included test cases in a run

7 Product Switcher

The Product Switcher lets you move between TraceBoard, TraceDocs, and TraceTest without logging in again.

Where to find it:

- The Product Switcher is located in the top-right corner of the navigation bar. It displays the name of the product you are currently using (e.g., "TraceBoard").



How to switch products:

- Click the product name button in the top-right corner of the navbar.
- A dropdown menu appears listing all three products:
 - Click the product you want to switch to.
 - The browser navigates to that product's URL. Your session is preserved across products.

Tips:

- The product you are currently on is highlighted in the dropdown (bold text, blue background).
- Deep links (e.g., a link to a specific test run) open the correct product automatically.
- You can also bookmark individual product URLs for direct access.

8 Administration

8.1 User Management

User management is available to users with the **Admin** role.

Adding a user:

1. Go to **Settings** → **Users** in TraceBoard.
2. Click **Add User**.
3. Enter the email address, name, and select a role.

Removing a user:

1. In the Users list, find the user and click the delete action.
2. Confirm the deletion. The user will no longer be able to log in.

Roles:

Role	Permissions
Admin	Full access: manage users, configure SSO, manage projects, create baselines, view everything.
Member	Standard access: create/edit items, create baselines, generate documents, execute tests.
Viewer	Read-only access: view items, baselines, documents, and test results. Cannot modify data.

Roles apply across all three products. An Admin in TraceBoard is also an Admin in TraceDocs and TraceTest.

Settings

Data Management
AI Import
Members
Compliance
License
Preferences

Project Members (2)

Name	Email	Role	Actions
Super Admin	admin@traceboard.local	Admin ▼	Reset Remove
Demo User	demo@traceboard.local	Member ▼	Reset Remove

Add Member

User

Role

Member ▼

Add

8.2 SSO (OIDC)

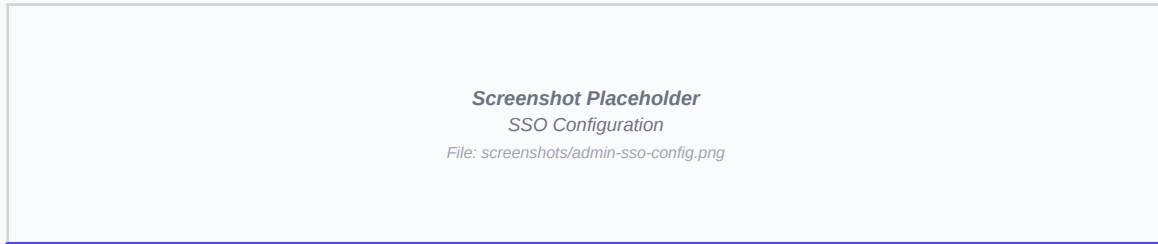
The TraceBoard Suite supports Single Sign-On via OpenID Connect (OIDC). This allows users to log in with their existing identity provider.

Supported providers:

- Keycloak
- Azure AD
- Any OIDC-compliant provider (Auth0, Okta, Google, etc.)

Configuration:

1. In TraceBoard, go to **Settings** → **SSO**.
2. Provide the following OIDC configuration details:
 1. Click **Enable OIDC**.
 2. Once configured, the login page shows an additional "Log in with SSO" button.



SSO applies across all three products. Users authenticated via SSO have their roles mapped from the identity provider's claims.

8.3 LLM Configuration

The "Ask AI" feature in TraceBoard (Section 3.7) relies on Ollama — a locally running LLM server.

Installing and configuring Ollama:

1. Install Ollama from ollama.com.
2. Pull a model:

```
bash
```

```
ollama pull llama3.2
```

```
`
```

1. Verify Ollama is running at `http://localhost:11434`.

Configuration in Docker Compose:

Set the following environment variables in your `docker-compose.yml` for the `traceboard-backend` service:

```
yaml
```

```
environment:
  OLLAMA_URL: http://host.docker.internal:11434
  LLM_MODEL: llama3.2
```

Using a cloud LLM provider:

As an alternative to local Ollama, you can use a cloud provider (e.g., Mistral AI) by setting:

```
yaml
```

```
environment:
  LLM_API_URL: https://api.mistral.ai/v1
  LLM_API_KEY: your-api-key
  LLM_MODEL: ministral-14b-2512
```

After configuration, restart the TraceBoard backend and the "Ask AI" button appears in the interface.

8.4 Backup and Restore

Database backup (PostgreSQL):

```
bash
```

```
docker compose exec db pg_dump -U traceboard traceboard > backup.sql
```

This creates a full SQL dump of the entire suite database (all product schemas).

Restore from backup:

```
bash
```

```
docker compose exec -T db psql -U traceboard traceboard < backup.sql
```

Important notes:

- Stop the application services before restoring to avoid data conflicts.
- The backup includes all three product schemas (`traceboard.*`, `tracedocs.*`, `tracetest.*`).
- Test your backups regularly by restoring to a staging environment.

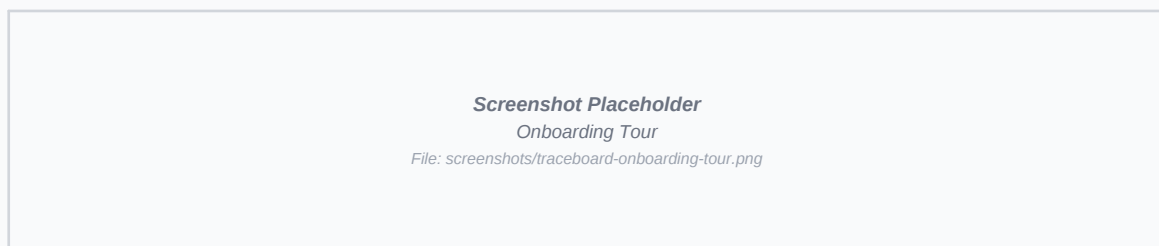
8.5 Onboarding Tour

TraceBoard Suite includes a guided onboarding tour that walks new users through the main features and interface.

Starting the tour:

1. The onboarding tour starts automatically when you log in for the first time.
2. To restart the tour at any time, go to **Settings** → **Help** and click **Start Onboarding Tour**.
3. The tour guides you step by step through:
 1. Each step highlights the relevant UI element and provides a short explanation. Click **Next** to advance, **Back** to revisit a step, or **Skip** to exit the tour.
 2. You can dismiss the tour at any time and restart it later from Settings.

Tip: The onboarding tour is a quick way to get team members productive. Point new users to this tour before they start working with the suite.



9 Troubleshooting

9.1 License Not Found or Expired

- Verify that `traceboard.license` is in the same directory as `docker-compose.yml`.
- Check the `expires_at` date in the license file — if it has passed, contact your sales representative.
- If the license file was recently updated, restart all services: `docker compose restart`.
- Check the logs for license-related errors: `docker compose logs traceboard-backend`.

9.2 TraceDocs or TraceTest Cannot Connect to TraceBoard

- Ensure TraceBoard is running and healthy: `docker compose ps` should show all services as up.
- Check that `TRACEBOARD_API_URL` is set correctly in the TraceDocs/TraceTest configuration.
- Check the logs: `docker compose logs tracedocs-backend` OR `docker compose logs tracetest-backend`.

9.3 LLM (AI) Connection Issues

If the "Ask AI" feature, Audit Narrative generation, or consistency checks are not responding, follow this structured guide.

Environment Variables (Required)

TraceBoard connects to an LLM via the OpenAI-compatible API. Ensure these variables are set in your `docker-compose.yml` (or `.env`):

Variable	Description	Example
<code>LLM_API_URL</code>	Base URL of the LLM API (must end with <code>/v1</code> for OpenAI-compatible endpoints).	<code>https://api.openai.com/v1</code> , <code>http://localhost:11434/v1</code> (Ollama), <code>http://vllm-server:8000/v1</code>
<code>LLM_API_KEY</code>	API key (or placeholder for local servers).	<code>sk-...</code> (OpenAI), <code>ollama</code> (Ollama), <code>dummy</code> (vLLM)
<code>LLM_MODEL</code>	Model name (must match what the provider expects).	<code>gpt-4o</code> , <code>claude-3-5-sonnet</code> , <code>gemini-1.5-pro</code> , <code>deepseek-v4</code> , <code>llama3.2</code>
<code>AI_CAPABILITY_LEVEL</code>	Optional override for feature availability (basic, medium, full, full+).	<code>full</code>

Step 1 – Verify the API Endpoint is Reachable

From within the TraceBoard container (or from the host if the container can reach it), test connectivity:

```
bash

# Test basic connectivity (no authentication)
curl -v http://localhost:11434/v1/models

# For cloud APIs, test with a simple request
curl https://api.openai.com/v1/models \
  -H "Authorization: Bearer sk-..."
```

If you get a connection timeout or refused, check:

- Network configuration (Docker networks, firewalls).
- For self-hosted LLMs (Ollama, vLLM), ensure the service is running and accessible from the TraceBoard container.
- For cloud APIs, verify your internet connection and any proxy settings.

Step 2 – Check API Key & Permissions

Cloud providers:

- **OpenAI:** `sk-...` – ensure the key is valid and has billing enabled.
- **Anthropic (Claude):** Use the OpenAI-compatible proxy (e.g., `https://api.anthropic.com/v1`). Requires an API key with appropriate model access.
- **Google Gemini:** Use the OpenAI-compatible endpoint (e.g., `https://generativelanguage.googleapis.com/v1beta/openai/`). Requires API key.
- **DeepSeek, Mistral, etc.:** Similar, ensure the key is active.

Self-hosted (Ollama, vLLM):

- **Ollama:** No API key required – use `ollama` or leave empty.
- **vLLM:** Usually no key required – use any string.

Test the key manually:

```
bash
```

```
curl https://api.openai.com/v1/chat/completions \
  -H "Authorization: Bearer sk-..." \
  -H "Content-Type: application/json" \
  -d '{"model": "gpt-4o", "messages": [{"role": "user", "content": "Hello"}]}'
```

If you get a 401/403, the key is invalid or lacks permissions.

Step 3 – Verify the Model Name

The `LLM_MODEL` must exactly match a model name supported by the provider.

- **Ollama:** `ollama list` to see installed models. Use the exact name (e.g., `llama3.2`).
- **vLLM:** The served model name is set at startup; check the server logs.
- **Cloud:** Use the exact model ID (e.g., `gpt-4o`, `claude-3-5-sonnet-20241022`, `gemini-1.5-pro`).

Step 4 – Check TraceBoard Logs

View the backend logs for LLM-related errors:

```
bash
```

```
docker compose logs traceboard-backend | grep -i "llm\|ai\|openai\|anthropic"
```

Common errors:

- `openai.APIError: 401 Unauthorized` – invalid API key.
- `openai.APIError: 404 Not Found` – wrong URL or model name.
- `openai.APITimeoutError` – network issues or server overload.
- `openai.RateLimitError` – quota exceeded (check billing).

Step 5 – Specific Scenarios

A) Ollama (local)

Ensure Ollama is running:

```
bash
```

```
curl http://localhost:11434/api/tags
```

Pull a model:

```
bash
```

```
ollama pull llama3.2
```

In `docker-compose.yml`, set:

```
yaml
```

```
LLM_API_URL=http://host.docker.internal:11434/v1
LLM_API_KEY=ollama
LLM_MODEL=llama3.2
```

B) vLLM (self-hosted, local network)

Ensure vLLM server is running:

```
bash
```

```
curl http://vllm-server:8000/v1/models
```

Set:

```
yaml
```

```
LLM_API_URL=http://vllm-server:8000/v1
LLM_API_KEY=dummy
LLM_MODEL=your-served-model
```

If vLLM is on a different host, use its IP/hostname.

C) Cloud Providers (OpenAI, Anthropic, Gemini)

OpenAI:

```
yaml
```

```
LLM_API_URL=https://api.openai.com/v1
LLM_API_KEY=sk-...
LLM_MODEL=gpt-4o
```

Anthropic (Claude) via OpenAI-compatible proxy (use the official proxy or a third-party adapter like LiteLLM):

```
yaml
```

```
LLM_API_URL=https://api.anthropic.com/v1
LLM_API_KEY=sk-ant-...
LLM_MODEL=claude-3-5-sonnet-20241022
```

Google Gemini (OpenAI-compatible):

```
yaml
```

```
LLM_API_URL=https://generativelanguage.googleapis.com/v1beta/openai
LLM_API_KEY=your-google-api-key
LLM_MODEL=gemini-1.5-pro
```

DeepSeek / Mistral / etc.: Use their respective OpenAI-compatible base URLs and model names.

Step 6 – Adjust AI Capability Level

If the LLM connection works but certain features (e.g., Audit Narrative) are not available, check `AI_CAPABILITY_LEVEL`.

- `basic` – only simple tasks.
- `medium` – most features.
- `full` – all features (default if unset).
- `full+` – experimental features.

Some models may not support all capabilities; test with `full`.

Step 7 – Restart the Backend

After any configuration change, restart:

```
bash
```

```
docker compose restart traceboard-backend
```

Still Not Working?

- Check the exact error message in the logs.
- Verify that your provider supports the `/chat/completions` endpoint (most do).
- If using a proxy or gateway (e.g., LiteLLM, Azure OpenAI), ensure the endpoint is correctly configured.
- For cloud providers, check billing and quota usage.

9.4 PDF Generation Fails

- Verify the selected baseline has requirements — an empty baseline produces an empty document (and may appear as a failure).
- Check TraceBoard API reachability from within the TraceDocs container:
- Large baselines with many requirements may take longer to generate. Wait and check the generation status in the History tab.
- Check the logs: `docker compose logs tracedocs-backend`.

9.5 Cannot Switch Products

- Ensure you are logged in. The Product Switcher requires an active session.
- If clicking a product in the dropdown does nothing, verify the product URLs are accessible:
- Try navigating directly to the product URL in a new tab.
- If a product returns an error, check its backend is running: `docker compose ps`.

10 Glossary

Term	Definition
Baseline	A point-in-time snapshot of all items and relations in a project. Immutable once created. Used as the source for document generation and test runs.

Term	Definition
Circular Reference	A trace cycle where items link back to themselves through a chain of relations (e.g., $A \rightarrow B \rightarrow C \rightarrow A$). Detected and flagged automatically.
Coverage	The percentage of requirements that have been verified by at least one passing test.
Evidence	Files (screenshots, logs) attached to a test result as proof of execution.
Flow View	An interactive visual graph showing items as nodes and relations as edges.
Gap	A requirement that has no linked tests (no <code>verifies</code> relation).
Generation	The process of creating a document (PDF or Word) from a baseline and a template in TraceDocs.
Grid View	A compliance matrix showing requirements against tests in a table format.
Item	A unit of trace data in TraceBoard. Can be a Requirement, Task, Bug, or Test.
Product Switcher	The dropdown in the top-right navbar that lets you switch between TraceBoard, TraceDocs, and TraceTest.
Relation	A directed link between two items (e.g., <code>verifies</code> , <code>implements</code> , <code>parent</code> , <code>relates-to</code>).
SRS	Software Requirements Specification — a compliance document listing all requirements.
Test Case	An item of type <code>test</code> in TraceBoard, linked to one or more requirements via <code>verifies</code> relations.
Test Run	A named execution session in TraceTest where test cases from a baseline are executed and results are recorded.
Trace	The network of items and relations that shows how requirements are implemented, verified, and decomposed.
Traceability	The mapping between requirements and test cases that shows which tests verify which requirements.

11 Support

- **Documentation:** See the Suite Architecture for technical details.

- **Product Specs:** TraceDocs Product Spec and TraceTest Product Spec for detailed feature descriptions.
- **Issue Tracker:** Report bugs or request features through the project's issue tracker.