Quathera

AI Form Mapping Guide

How AI analyzes and maps every field, condition, and relationship in your forms.

Version 1.0 | 2025

Table of Contents

1. What Is Form Mapping?

Form Mapping is the deep analysis phase that happens after discovery. While discovery finds WHERE forms are, mapping understands WHAT each form contains and HOW it works.

Discovery vs Mapping:

- Discovery: Finds form pages and navigation paths
- Mapping: Analyzes every field, condition, and behavior

Why Mapping Matters:

Mapping provides the detailed understanding needed to generate comprehensive test scenarios. Without mapping, AI wouldn't know:

- What fields exist and their types
- Which fields are required vs optional
- How fields interact (conditional logic)
- What validation rules apply
- How to verify data was saved correctly

2. Starting the Mapping Process

To map a discovered form:

Steps:

1. Go to 'Form Pages Discovery' in the sidebar
2. Find the form you want to map in the list
3. Click the Map icon (■■) in the Actions column
4. AI will launch a browser and begin analysis
5. Watch as AI fills fields, clicks buttons, and explores paths
6. Mapping completes automatically

Mapping Time:

• Simple forms (5-10 fields): 2-5 minutes

• Medium forms (15-30 fields): 5-15 minutes

• Complex forms (50+ fields, multiple paths): 15-45 minutes

3. What AI Discovers

During mapping, AI performs comprehensive analysis:

All Form Fields:

Every input field is identified with its type, label, and locator.

Field Properties:

- Required vs optional
- Default values
- Placeholder text
- Max length constraints
- Format validation (email, phone, etc.)

Conditional Logic:

Fields that appear/disappear based on other field values.

Form Actions:

- Submit/Save buttons
- Cancel buttons
- Next/Previous (for wizards)

Verification Points:

Where and how to verify data after saving (grid columns, detail views).

4. Field Types Supported

AI recognizes and handles all common field types:

Text Inputs:

- Single-line text
- Email fields
- Password fields
- Phone/number fields
- URL fields

Multi-line:

- Textareas
- Rich text editors (WYSIWYG)

Selection:

- Dropdowns (select boxes)
- Multi-select
- Autocomplete/typeahead
- Radio buttons
- Checkboxes

Date/Time:

- Date pickers
- Time pickers
- DateTime pickers

Other:

- File uploads

- Color pickers
- Sliders/ranges
- Toggle switches

5. Junction Paths (Conditional Logic)

Many forms have conditional fields that appear based on other selections. Quathera calls these 'Junctions'.

Example:

An Employee form with an 'Employment Type' dropdown:

- If 'Full-Time' is selected: Salary and Benefits fields appear
- If 'Part-Time' is selected: Hourly Rate field appears
- If 'Contractor' is selected: Company Name and Contract End Date appear

How AI Handles Junctions:

1. AI identifies the junction field (e.g., Employment Type dropdown)
2. AI tries each option and records which fields appear
3. AI creates separate 'paths' for each option
4. Each path is mapped completely

Result:

For the example above, AI would create 3 paths:

- Path 1: Full-Time (with Salary, Benefits fields)
- Path 2: Part-Time (with Hourly Rate field)
- Path 3: Contractor (with Company, Contract End fields)

Test scenarios are generated for ALL paths, ensuring complete coverage.

6. Parent-Child Relationships

AI automatically detects when forms depend on each other:

How AI Detects Relationships:

- Dropdown fields that reference other entities
- Forms that are only accessible from a parent record
- URL patterns indicating hierarchy (e.g., /customer/123/orders)

Example Hierarchy:

Department (Root)

- ■■■ Employee (Child of Department)
- ■■■ Task (Child of Employee)
- ■■■ Timesheet (Child of Task)

What This Means for Testing:

To test a Timesheet, AI must:

1. Create a Department first
2. Create an Employee in that Department
3. Create a Task for that Employee
4. Finally create the Timesheet for that Task

AI handles this automatically when generating test scenarios.

7. Multi-User Workflows

Some forms require different users for different actions:

Example:

- Admin creates a Department
- HR creates Employees in the Department
- Manager assigns Tasks to Employees
- Employee creates Timesheets for Tasks

How AI Handles This:

During mapping, AI records which user type was logged in when discovering each form. This information is used to generate test scenarios with proper login/logout cycles.

Automatic Login Cycling:

When running tests, AI automatically:

1. Logs in as Admin to create Department
2. Logs out
3. Logs in as HR to create Employee
4. Logs out
5. And so on...

8. Spec vs Discovery Comparison

If you provide a specification document, AI compares it against what was discovered:

What Gets Compared:

- Fields listed in spec vs fields found in form
- Field types (spec says dropdown, form has text input)
- Required/optional status
- Validation rules

Discrepancies Flagged:

- Field in spec but missing from form
- Field in form but not in spec (undocumented)
- Type mismatch between spec and implementation
- Validation differences

Why This Matters:

Spec drift is a common problem. Developers add fields without updating docs, or specs describe features not yet implemented. This comparison catches these issues early.

9. Reviewing Mapping Results

After mapping completes, you can review:

Field List:

All discovered fields with their types, locators, and properties.

Junction Paths:

All conditional paths through the form.

Relationships:

Parent forms this form depends on.

Verification Points:

Where to check values after save (grid columns, detail view fields).

Spec Comparison:

Any discrepancies between spec and discovered form.

Next Step:

Once mapping is complete, AI can automatically generate test scenarios for this form. See the Test Scenarios Guide for details.