Requirements Traceability Matrix

JSONPlaceholder API Testing Project

Date: August 9, 2025

Author: Victor Murashev

1. Overview

1.1 Purpose

This Requirements Traceability Matrix (RTM) ensures complete test coverage by mapping business requirements to test cases and tracking the relationship between requirements, test cases, and defects.

1.2 Traceability Types

- **Forward Traceability:** Requirements → Test Cases
- **Backward Traceability:** Test Cases → Requirements
- **Bidirectional Traceability:** Requirements ↔ Test Cases ↔ Defects

2. Requirements Coverage Matrix

2.1 Posts API Requirements

Req ID	Requirement Description	Priority	Test Case IDs	Test Status	Coverage %	Comments
FR-001	**Posts Retrieval**					
FR-001.1	Get all posts via GET /posts	High	TC-001, TC-002	✅ Passed	100%	Smoke + Functional
FR-001.2	Get single post via GET /posts/{id}	High	TC-003, TC-004, TC-005	✅ Passed	100%	Valid + Invalid IDs
FR-001.3	Handle non-existent post ID	Medium	TC-006	✅ Passed	100%	Error handling
FR-002	**Posts Creation**					
FR-002.1	Create post via POST /posts	High	TC-007, TC-008	✅ Passed	100%	Valid data

FR-002.2	Validate required fields (title, body, userId)	High	TC-009, TC-010, TC-011	✅ Passed	100%	Field validation
FR-002.3	Handle invalid user ID in post creation	Medium	TC-012	✅ Passed	100%	Data validation
FR-003	**Posts Update**					
FR-003.1	Update post via PUT /posts/{id}	High	TC-013, TC-014	✅ Passed	100%	Full update
FR-003.2	Partial update via PATCH /posts/{id}	Medium	TC-015	✅ Passed	100%	Partial update
FR-003.3	Handle update of non-existent post	Medium	TC-016	✅ Passed	100%	Error handling
FR-004	**Posts Deletion**					
FR-004.1	Delete post via DELETE /posts/{id}	High	TC-017	✅ Passed	100%	Successful deletion
FR-004.2	Handle deletion of non-existent post	Medium	TC-018	✅ Passed	100%	Error handling

2.2 Users API Requirements

Req ID	Requirement Description	Priority	Test Case IDs	Test Status	Coverage %	Comments
FR-005	**Users Retrieval**					
FR-005.1	Get all users via GET /users	High	TC-019, TC-020	✅ Passed	100%	Smoke + Functional
FR-005.2	Get single user via GET /users/{id}	High	TC-021, TC-022, TC-023	✅ Passed	100%	Valid + Invalid IDs
FR-005.3	Handle non-existent user ID	Medium	TC-024	✅ Passed	100%	Error handling
FR-006	**Users Creation**					
FR-006.1	Create user via POST /users	High	TC-025, TC-026	✅ Passed	100%	Valid data
FR-006.2	Validate required fields (name, username, email)	High	TC-027, TC-028, TC-029	✅ Passed	100%	Field validation
FR-006.3	Validate email format	Medium	TC-030	✅ Passed	100%	Email validation
FR-007	**Users Update**					
FR-007.1	Update user via PUT /users/{id}	High	TC-031, TC-032	✅ Passed	100%	Full update
FR-007.2	Partial update via PATCH /users/{id}	Medium	TC-033	✅ Passed	100%	Partial update

FR-007.3	Handle update of non-existent user	Medium	TC-034	✅ Passed	100%	Error handling
FR-008	**Users Deletion**					
FR-008.1	Delete user via DELETE /users/{id}	High	TC-035	✅ Passed	100%	Successful deletion
FR-008.2	Handle deletion of non-existent user	Medium	TC-036	✅ Passed	100%	Error handling

2.3 Non-Functional Requirements

Req ID	Requirement Description	Priority	Test Case IDs	Test Status	Coverage %	Comments
NFR-001	**Performance**					
NFR-001.1	Response time < 2 seconds	High	TC-037, TC-038	✅ Passed	100%	All endpoints
NFR-001.2	API availability > 99%	Medium	TC-039	⏳ In Progress	50%	Monitoring setup
NFR-002	**Data Integrity**					
NFR-002.1	Response data matches schema	High	TC-040, TC-041	✅ Passed	100%	JSON schema validation
NFR-002.2	Consistent data types	High	TC-042	✅ Passed	100%	Type validation
NFR-003	**Error Handling**					
NFR-003.1	Proper HTTP status codes	High	TC-043, TC-044	✅ Passed	100%	Status code validation
NFR-003.2	Meaningful error messages	Medium	TC-045	✅ Passed	100%	Error response validation

3. Test Case Mapping

3.1 Test Cases by Priority

Priority	Total Requirements	Covered	Coverage %	Status
High	16	16	100%	✅ Complete
Medium	12	11	92%	⏳ In Progress
Low	0	0	N/A	-
Total	**28**	**27**	**96%**	✅ On Track

3.2 Test Cases by Feature

Feature	Total Test Cases	Passed	Failed	Not Run	Success Rate
Posts API	18	18	0	0	100%
Users API	18	18	0	0	100%
Performance	3	2	0	1	67%
Data Validation	6	6	0	0	100%

Total	**45**	**44**	**0**	**1**	**98%**

4. Defect Traceability

4.1 Defects by Requirement

Defect ID	Requirement ID	Severity	Status	Description	Test Case
-	-	-	-	No defects found	-

4.2 Defect Summary

Severity	Open	Resolved	Total
Critical	0	0	0
High	0	0	0
Medium	0	0	0
Low	0	0	0
Total	**0**	**0**	**0**

5. Gap Analysis

5.1 Uncovered Requirements

Req ID	Description	Reason	Action Plan
NFR-001.2	API availability monitoring	Environment limitation	Implement in Sprint 2

5.2 Test Cases Without Requirements

Test Case ID	Description	Action Required
-	All test cases mapped to requirements	None

6. Coverage Metrics

6.1 Overall Coverage

```
Requirements Coverage: 96% (27/28)
Test Execution: 98% (44/45)
Defect Density: 0 defects/requirement
```

6.2 Coverage by Test Type

Test Type	Coverage	Status
Functional	100%	✅ Complete
Validation	100%	✅ Complete
Performance	67%	⏳ In Progress
Error Handling	100%	✅ Complete

7. Traceability Reports

7.1 Forward Traceability

```
28 Requirements → 45 Test Cases
Coverage: 96%
```

7.2 Backward Traceability

```
45 Test Cases → 28 Requirements
All test cases linked to requirements
```

7.3 Bidirectional Traceability

```
Requirements ↔ Test Cases: 96% coverage
Test Cases ↔ Defects: 0% (no defects)
Requirements ↔ Defects: 0% (no defects)
```