

Functional Specification Document (FSD)

Project Name: Publishing Details Form

Author: Ahmad Warsi

Date: 08-Dec-2025

Version: 1.0

1. Introduction

1.1 Purpose:

The purpose of this document is to define the functional requirements for the **Publishing Details Form**, which allows users to capture API details and download them in Excel format.

1.2 Scope:

This project provides:

- Dynamic form creation for multiple API entries.
- Validation for fields like API Name (camelCase), JIRA ID, mandatory selections.
- Auto-population and freezing of fields based on Category selection (e.g., "Outbound").
- Risk Score calculation and classification.
- Excel export of all API entry data.

2. Functional Requirements

Feature	Description	Behavior / Rules
Dynamic API Entries	Users can add or delete API entries	At least one entry must remain.
API Name Validation	Ensures API name is camelCase	Regex check: <code>/^[a-z]+(-[a-zA-Z]+)*(-v[0-9]+)?\$/</code>
Category Handling	Auto-fill GAS ID and Risk Classification for "Outbound"	- GAS ID = "NA/Outbound" - Risk Score cleared & disabled - Risk Classification = "Outbound" - GAS ID disabled
Risk Score / Classification	Automatic classification based on score	0-40 → Low, 41-65 → Medium, 66+ → High
JIRA ID Validation	Must start with "APIGATE"	Alert if invalid
Excel Export	Download all entries in Excel	Includes all fields in header row

Feature	Description	Behavior / Rules
File Upload Preview	Workflow image preview before upload	Shows preview if image selected
Mandatory Fields	All fields validated before download	Alerts user if any field is empty

3. Non-Functional Requirements

Requirement	Description
Browser Compatibility	Chrome, Firefox, Edge
Responsiveness	Form layout adjusts for desktop and tablet
Performance	Excel download should be < 2s for 50 entries
Security	No sensitive data stored client-side
Technology Stack	HTML, CSS, JavaScript, XLSX.js library

4. UI/UX Requirements

- Form Layout: Card-based for each API entry.
- Buttons: Add Entry, Delete Entry, Download Excel.
- Field Validations: Inline alerts for invalid inputs.
- Disabled Fields: GAS ID, Risk Score, Risk Classification when Category = Outbound.
- Preview: Workflow image preview shown below file input.

5. Data Fields

Field	Type	Validation / Notes
API Name on Gateway	Text	camelCase only
Display Name	Text	Mandatory
Short Description	Textarea	Mandatory
API Product Description	Textarea	Mandatory
API Product Features	Textarea	Optional
Category	Dropdown	Mandatory; triggers field behavior

Field	Type	Validation / Notes
Workflow File	File	Optional; image preview
Use Cases	Textarea	Optional
API Product Name	Text	Optional
Change Ticket ID	Text	Optional
Data Classification	Dropdown	Mandatory
Data Classification Details	Textarea	Optional
Description	Text	Optional
Exception Taken	Dropdown	Optional
Exception Reason	Textarea	Optional
GAS ID	Text	Auto-populated if Outbound
JIRA ID	Text	Must start with APIGATE
API Offering	Dropdown	Mandatory
Middleware	Dropdown	Mandatory
Orchestrated/Pass Through	Dropdown	Mandatory
API Reusability	Dropdown	Optional
Producer	Text	Optional
Producer End Point	Text	Optional
Security Test Status	Dropdown	Optional
Hosting Date UAT/PROD	Date	Optional
API Risk Classification	Dropdown	Auto-calculated or Outbound
Risk Score	Number	Disabled for Outbound; auto-classified

6. Workflows

6.1 Add New API Entry

- Click Add New API Entry → A new card appears.
- Fields reset for input.
- Category rules applied dynamically.

6.2 Delete API Entry

- Click Delete This Entry → Card removed.
- Minimum one card must remain.

6.3 Category Selection

- Selecting Outbound disables Risk Score & GAS ID and sets Risk Classification.
- Other categories enable fields for input.

6.4 Risk Classification Calculation

- User inputs Risk Score → Classification updated automatically.

6.5 Download Excel

- Click Download Excel → Form validated → Excel file downloaded with all data.
-

7. Assumptions

- Users have modern browsers with JS enabled.
 - File upload previews only images (png, jpg, jpeg).
 - Excel download handled via client-side XLSX.js library.
-

8. Future Enhancements

- API-based form submission to backend.
- Role-based access for edit/view.
- Multi-user collaboration.